(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3836928号
(P3836928)
(45)発行日　平成18年10月25日(2006.10.25)　(24)登録日　平成18年8月4日(2006.8.4)

(51)Int.Cl.　F I
G06F 12/00 (2006.01)　G06F 12/00 513D
G06F 12/00 547Q

請求項の数 5　(全 23 頁)

(21)出願番号　特願平9-41906
(22)出願日　平成9年2月26日(1997.2.26)
(65)公開番号　特開平10-240588
(43)公開日　平成10年9月11日(1998.9.11)
審査請求日　平成15年1月17日(2003.1.17)

(73)特許権者　000005108
株式会社日立製作所
東京都千代田区丸の内一丁目６番６号
(74)代理人　100083552
弁理士　秋田　収喜
(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　情報・通信開発本部内
(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　情報・通信開発本部内
(72)発明者　鳥居　俊一
神奈川県川崎市幸区鹿島田８９０番地　株式会社日立製作所　情報・通信開発本部内

最終頁に続く

(54)【発明の名称】データベース処理方法

(57)【特許請求の範囲】
【請求項１】
データベースに予め定義登録しておいた関数を処理要求に応じて起動してデータベース処理を行うデータベース処理方法において、
特定のユーザ定義関数とデータベース領域のデータにインデクスアクセスを行う際に用いられる特定のインデクスとを対応付けるインデクス定義情報を作成し、
データベース領域のデータにインデクスアクセスを行う際に用いられる特定のインデクスとデータベース領域のデータに特定の処理を行う実現モジュールとを対応付けるモジュール定義情報を作成し、
データベースへの処理要求中のユーザ定義関数が前記インデクス定義情報により特定のインデクスに対応付けられている場合に、前記特定のインデクスを用いたアクセスを行う特定の実現モジュールを前記モジュール定義情報から選択し、前記選択した実現モジュールを用いてデータベース処理を行うことを特徴とするデータベース処理方法。
【請求項２】
複数の異なるユーザ定義関数と特定のインデクスとを前記インデクス定義情報で対応付けることにより、データベース領域のデータに特定の処理を行う実現モジュールを複数の異なるユーザ定義関数で共有することを特徴とする請求項１に記載されたデータベース処理方法。
【請求項３】
前記インデクス定義情報中のインデクスと、前記モジュール定義情報中の当該インデクス

及び当該インデクスに対応付けられた実現モジュールを変更することにより、特定のユーザ定義関数に対するデータベース処理の内容を変更することを特徴とする請求項１または請求項２のいずれかに記載されたデータベース処理方法。
【請求項４】
前記データベースへの処理要求が更新要求である場合に、前記モジュール定義情報で前記特定のユーザ定義関数に対応付けられたインデクスキーを生成する実現モジュールによりインデクスキーを生成し、前記生成したインデクスキーによりインデクスを更新することを特徴とする請求項１乃至請求項３のいずれか１項に記載されたデータベース処理方法。
【請求項５】
前記データベースへの処理要求が更新要求である場合に、更新用のデータをインデクス更新用キーに指定してインデクスを更新することを特徴とする請求項１乃至請求項４のいずれか１項に記載されたデータベース処理方法。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は、データベースを処理するデータベース処理方法に関し、特に、ユーザがデータ型及びその振る舞いを定義することが可能なユーザ定義関数を用いてインデクスアクセスを行うデータベース処理方法に適用して有効な技術に関するものである。
【０００２】
【従来の技術】
現在、データベース言語ＳＱＬではＳＱＬ３（ＩＳＯ、ＡＮＳＩ）の規格化が進められており、そのＳＱＬ３の主要機能の１つにＡＤＴ（Ａｂｓｔｒａｃｔ　Ｄａｔａ　Ｔｙｐｅ、抽象データ型）がある。
【０００３】
ＡＤＴとは、ユーザ定義のデータ型であり、オブジェクト指向の概念を取り入れ、ＡＤＴデータに対する操作もメソッド（関数、手続き）としてユーザが定義するものである。ＡＤＴを用いると、複雑なデータ構造を実現することができる。また、ＡＤＴデータの値自身の振る舞い（付随する機能）をＡＤＴ関数として定義することができる。
【０００４】
ＡＤＴの定義では、値を表現する為の属性（群）の仕様と、その振る舞いを実現する一群の操作をＡＤＴ関数として規定する。これらＡＤＴ定義情報は、表定義等と同様にディクショナリ情報としてデータベース管理システムにおいて管理されるのが実装として一般的である。
【０００５】
ＡＤＴ関数の定義には、ＳＱＬ自身や、Ｃ言語等の一般プログラミング言語で記述してコンパイルし、システムに登録したモジュールを指定することもでき、これらＡＤＴ関数の内部的実現形態であるモジュールとＡＤＴ関数との関連は、ＡＤＴを定義する為のＡＤＴ定義文（ＣＲＥＡＴＥ　ＴＹＰＥ文）に記述される。
【０００６】
ＡＤＴを使うことによって、マルチメディアデータに対応する機能をデータベース管理システムの機能として実現することができる様になる。これは、今までアプリケーションプログラムで行っていた処理を、データベースシステム側で高速にかつ低開発コストで実現できることを意味している。
【０００７】
一方、一般の検索システムでは、ある特有の機能を高速に実現する為に、その機能特有のアクセスメソッドを利用することが多く、その代表的なものとしてインデクスが挙げられる。
【０００８】
例えば、全文検索システムでは、文書を高速に検索する為の抽出キーワードや文字成分表等で実現される専用インデクスが使われている。また、多くのデータベース管理システム

では、システムに内蔵されているＢ木インデクス等のアクセスメソッドを使用することによりユーザ問合わせ処理（検索処理）の高速化を図っている。

【０００９】

ＡＤＴデータに関しても、あるＡＤＴ特有の評価（検索）や高速な評価（検索）を実現する為の特有のインデクスの利用が考えられるが、ＡＤＴの恩恵を十分に享受する為には、ユーザ定義のアクセスメソッドすなわちインデクスを組込む仕掛けがデータベース管理システムに必要である。

【００１０】

【発明が解決しようとする課題】

ところで、従来のデータベース処理方法では、ユーザ定義インデクスを組込む一つの方法として、ＡＤＴ関数を実現するモジュールを用いて関数内で検索を高速に実現するアクセスメソッドを内蔵する方法が挙げられるが、その場合、ＡＤＴ関数を実現するモジュールがユーザ定義インデクスをアクセスする機能を有する為、そのユーザ定義インデクスはＡＤＴに括り付けになり、ＡＤＴに対して独立性を失うことになる。

【００１１】

従って、前記従来のデータベース処理方法において、ＡＤＴ関数にユーザ定義インデクスをアクセスする機能を内蔵すると、いくら有用で汎用的に設計されたインデクスでも他のＡＤＴからは利用することができないという問題が生じていた。

【００１２】

また、前記従来のデータベース管理システム内に、Ｂ木インデクスだけでなく、文書検索用インデクスや画像検索用インデクス等あらゆるデータに対応したインデクスを実装することは不可能であり、多くのデータに対応したインデクスを実装するとデータベース管理システムが大きくなって、かえって不要なインデクスモジュールが内蔵されてしまうという問題があった。

【００１３】

本発明の目的は、特定のユーザ定義関数に依存しない汎用的なデータベース処理を行うことが可能な技術を提供することにある。

【００１４】

本発明の他の目的は、特定のインデクスアクセスを行う実現モジュールをユーザ定義関数から独立させることが可能な技術を提供することにある。

【００１５】

本発明の他の目的は、必要な実現モジュールのみを内蔵することが可能な技術を提供することにある。

【００１６】

本発明の前記並びにその他の目的と新規な特徴は、本明細書の記述及び添付図面によって明かになるであろう。

【００１７】

【課題を解決するための手段】

本願によって開示される発明のうち、代表的なものの概要を簡単に説明すれば、下記のとおりである。

【００１８】

すなわち、データベースに予め定義登録しておいた関数を処理要求に応じて起動してデータベース処理を行うデータベース処理方法において、データベースへの処理要求中のユーザ定義関数が前記インデクス定義情報により特定のインデクスに対応付けられている場合に、前記特定のインデクスを用いたアクセスを行う特定の実現モジュールを前記モジュール定義情報から選択し、前記選択した実現モジュールを用いてデータベース処理を行うものである。

【００１９】

以上の様に、前記データベース処理方法によれば、処理要求中のユーザ定義関数がインデクス定義情報に定義されたものである場合に、前記ユーザ定義関数に対応する実現モジュ

ールを実行してインデクスアクセスを行うので、特定のユーザ定義関数に依存しない汎用的なデータベース処理を行うことが可能である。

【００２０】

【発明の実施の形態】

以下に、本発明のデータベース処理方法において、データベースに予め定義登録しておいたユーザ定義関数を処理要求に応じて起動してデータベース処理を行う一実施形態のデータベース処理方法を実施する実施装置について説明する。

【００２１】

図１は、本実施形態のデータベース処理方法を実施する実施装置の概略構成を示す図である。図１において、１はデータベース管理システム、２はユーザ要求、３はモジュール定義情報、４はデータベース領域、５はディクショナリ、１０は問合せ解析処理部、２０はデータベース演算処理部、３０はユーザ定義インデクス管理部、３１はユーザ定義インデクス検索処理制御部、３２はユーザ定義インデクス更新処理制御部、４０はテーブルデータ管理部、４１はユーザ定義インデクス、４２はテーブルデータ、５０はモジュール起動部、５１はインデクスタイプ定義情報、５２はインデクス定義情報、５３はデータタイプ定義情報、５４はモジュール定義情報、６０はデータベース領域アクセス処理部、７０はモジュール定義情報登録部、８０は定義処理部、９０はディクショナリ管理部、１００ａ～１００ｄは各種モジュールである。

【００２２】

図１に示す様に、本実施形態のデータベース処理方法を実施する実施装置では、以下の様にユーザ定義インデクス４１の組込み及びアクセスを実現する。ここで、ユーザが新たにデータベース管理システム１に追加しようとするインデクスの種別を、以下インデクスタイプと呼ぶ。

【００２３】

本実施形態のデータベース処理方法を実施する実施装置では、あるインデクスタイプを適用したインデクスに対する検索、更新等を実現するモジュールは、ＡＤＴと同様にユーザ定義関数の実現モジュールとして、ユーザによりデータベース管理システム１へ組込まれる。図１の各種モジュール１００ａ～１００ｄが、組込まれたユーザ定義関数実現モジュールである。

【００２４】

まず、インデクスを新たにデータベース管理システム１に追加するユーザは、データベース管理システム１がそのインデクスタイプを認知し、そのインデクスタイプの実現モジュールを実行（起動）する為の情報をインデクスタイプ定義情報５１としてデータベース管理システム１に登録する。図１の例ではユーザ要求２のインデクスタイプ定義要求がそれにあたる。

【００２５】

次に、既にデータベースのデータを格納したテーブルの一つまたは複数のデータ項目に対して、あるインデクスタイプのインデクスを作成することにより、ユーザからの検索要求に対してそのインデクスの使用が可能となる。そのインデクス作成指示は、テーブル定義と同様にユーザがデータベース管理システム１に対して行う。図１の例ではユーザ要求２のインデクス定義要求がそれにあたる。

【００２６】

作成されたインデクスは、問合せ解析処理部１０にてユーザからの検索要求に応じて最適なアクセス経路として選定される。そして、ユーザ定義インデクス管理部３０の制御の元にアクセスされ高速または高機能検索を実現する。

【００２７】

インデクスタイプ定義要求には、インデクスタイプ名称やインデクスを維持する為の入力データ（以下そのデータをインデクスキーと呼ぶ）のデータ型等についての情報が含まれる。インデクス定義要求には、どのインデクスタイプのインデクスを作成するか、インデクスを作成及び維持するにはどのテーブルデータ４２を必要とするか、そのインデクスを

使用するのはどの様な場合か、等に関する情報が含まれる。これらの定義情報は、ディクショナリ管理部９０が管理するディクショナリ５に格納される。

【００２８】

次に図１を用い、本実施形態のデータベース処理方法を実施する実施装置の構成について詳細に説明する。

【００２９】

本実施形態のデータベース処理方法を実施する実施装置は、ユーザからのデータベース問合せ要求であるＳＱＬ（構造化照会言語）を受け取り、構文解析、意味解析処理を通してデータベースアクセスの最適なアクセス経路を決定する最適化処理を行い、決定したアクセス経路に基づいてデータベース処理用の内部処理コードを生成する問合せ解析処理部１０、

生成された内部処理コードを基にデータベースアクセスの制御を行うデータベース演算処理部２０、データベース演算処理部２０の要求指示により、ユーザ定義インデクス４１へのアクセスを行う為のモジュールの起動制御を行うユーザ定義インデクス管理部３０、

データベース演算処理部２０の要求指示により、テーブルデータ４２へのアクセス制御を行うテーブルデータ管理部４０、データベース演算処理部２０またはユーザ定義インデクス管理部３０からの要求指示により各種モジュール１００ａ～１００ｄを起動するモジュール起動部５０、

ユーザ定義インデクス４１及びテーブルデータ４２が格納されているデータベース領域４へのアクセスを行うデータベース領域アクセス処理部６０、ユーザから入力されるＡＤＴ関数またはユーザ定義インデクス４１へのアクセスを実現する為のモジュールに関するモジュール定義情報３を受け付け解析し、ディクショナリ５への登録を要求するモジュール定義情報登録部７０、

ユーザ要求２が各種定義要求の場合、問合せ解析処理部１０の解析結果に基づきディクショナリ５に対する登録或いは削除を要求する定義処理部８０、ディクショナリ５に対する登録処理、参照処理或いは削除処理を行うディクショナリ管理部９０で構成される。

【００３０】

ディクショナリ５には、テーブルやインデクスに関する定義情報等の各種定義情報が格納されている。そのディクショナリ５に格納されている定義情報には、インデクスタイプ定義情報５１、インデクス定義情報５２、データタイプ定義情報５３、モジュール定義情報５４が含まれる。

【００３１】

インデクスタイプ定義情報５１は、インデクスタイプ定義要求により入力された情報である。インデクス定義情報５２は、インデクスを作成する為のインデクス定義要求により入力された情報である。

【００３２】

データタイプ定義情報５３は、ＡＤＴに関する定義情報、すなわちＡＤＴを構成するデータ型及びそのＡＤＴの振る舞いを実現する為のＡＤＴ関数に関する情報である。そのＡＤＴ関数の中には、ユーザ定義インデクス４１を用いて実現されるものもある。データタイプ定義情報５３は、ユーザ要求２のデータタイプ定義要求によりユーザから入力されたものである。

【００３３】

そして、モジュール定義情報５４は、ＡＤＴ関数の実現またはユーザ定義インデクス４１の実現の為のモジュールがどの契機で起動されるかを示す情報である。

【００３４】

それらユーザ登録モジュールには、ＡＤＴ関数を実現する為のモジュール、ユーザ定義インデクス４１に対する検索処理を実現する為のモジュール、ユーザ定義インデクス４１に対する更新（登録、更新、削除）処理を実現する為のモジュール、ユーザ定義インデクス４１を更新する為の入力キーを生成する為のモジュール等がある。図１の各種モジュール１００ａ～１００ｄがそれらのモジュールにあたる。

【００３５】
図２は、本実施形態のデータベース処理方法を実施する実施装置のハードウェア構成の一例を示す図である。図２において、１０００はコンピュータシステム、１００１は主記憶装置、１００２はＣＰＵ、１００３は外部記憶装置、１００４は端末、１１００は処理プログラムである。
【００３６】
図２に示す様に、本実施形態のデータベース処理方法を実施する実施装置のハードウェア構成では、コンピュータシステム１０００は、ＣＰＵ１００２、主記憶装置１００１、磁気ディスク等の外部記憶装置１００３及び多数の端末１００４で構成される。
【００３７】
主記憶装置１００１上には、図１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１００３上にはデータベース管理システム１が管理するデータベースに関する各種定義情報を含むディクショナリ５と、定義されたテーブルデータ４２及びユーザ定義インデクス４１を含むデータベース領域４とが格納される。また、コンピュータシステム１０００にデータベース管理システム１を実現させる為の処理プログラム１１００も外部記憶装置１００３上に格納される。
【００３８】
図３は、本実施形態のデータベース処理方法を実施する実施装置のインデクス追加作成手順の一例を示す図である。図３において、３０１～３０６は手順である。
【００３９】
図３に示す様に、本実施形態のデータベース処理方法を実施する実施装置のインデクス追加作成手順では、ユーザがデータベース管理システム１に対して新たなインデクスを追加作成する際の作成手順を表しており、その概略は以下の様になる。
【００４０】
（１）インデクスタイプ定義（手順３０１）
（２）インデクス実現モジュール定義情報登録（手順３０２）
（３）データタイプ定義（手順３０３）
（４）ＡＤＴ実現モジュール定義情報登録（手順３０４）
（５）テーブル定義（手順３０５）
（６）インデクス定義（手順３０６）
まず、インデクスタイプ定義（手順３０１）において、追加しようとするインデクスタイプに関する情報を登録する。インデクスタイプ定義では、次の２つの項目を指定する。
【００４１】
（Ａ）インデクスを作成する際、どのタイプのインデクスかを指定する為の識別子として用いるインデクスタイプ名称。
【００４２】
（Ｂ）テーブルデータ４２の挿入、更新、削除操作に伴い、インデクスを更新する際の更新処理に入力として必要なデータ（インデクスキー）のデータ項目数及びそのデータ型。
【００４３】
登録の際のインタフェースとして、図３の例で示してある「ＣＲＥＡＴＥ　ＩＮＤＥＸ　ＴＹＰＥ文」の様なＳＱＬ形式に近い記述形式が考えられる。以下に、「ＣＲＥＡＴＥ　ＩＮＤＥＸ　ＴＹＰＥ文」の構文記述例を変形ＢＮＦ記法（バッカス記法）で示す。
【００４４】

```
CREATE INDEX TYPE インデクスタイプ名称
  [ FOR { データ型名 [、データ型名、…]
        |ADT型名
        |ADT型名 WITH 属性名 [、属性名、…] }
  ]
```

上記構文記述例では、ＦＯＲ句によって（Ｂ）のインデクスキーのデータ型及びそのデータ項目数（データ型の列挙数）を指定する。またインデクスキーとして、以下の型のデータを更新する際の入力とするということを指示している。

【００４５】

（ａ）一つまたは複数の基本データ型

（ｂ）ＡＤＴ型名で示されるＡＤＴ

（ｃ）ＡＤＴ型名で示されるＡＤＴデータ内の一つまたは複数の属性のデータそのインデクスキーのデータ項目数は、上記構文記述例のデータ型名またはＷＩＴＨ句の後の属性名の数で表される。

【００４６】

図３の「ＣＲＥＡＴＥ　ＩＮＤＥＸ　ＴＹＰＥ文」では、インデクスタイプ名称ＤＯＣ－ＩＮＤＥＸ、インデクスキーはＶＡＲＣＨＡＲ型１つ、というインデクスタイプ定義がなされている。

【００４７】

インデクスタイプ定義の手順３０１に続き、手順３０２において前記インデクスを実現するモジュールのモジュール定義情報５４の登録を行う。以下に図３のインデクスタイプ定義に基づいたインデクス実現モジュールのモジュール定義情報５４の登録の際のユーザインタフェースの一例を挙げる。

【００４８】

```
udimodule {
  インデクスタイプ名:DOC－INDEX、
  _p_doc_insert{
     起動契機    : AS_INDEX_INSERT
  }
  _p_doc_delete{
     起動契機    : AS_INDEX_DELETE
  }
  _p_doc_update
     起動契機    : AS_INDEX_UPDATE
  }
  _p_doc_scan{
     起動契機    : AS_INDEX_SCAN
  }
}
```

まず、ｕｄｉｍｏｄｕｌｅは、ユーザによって入力されたモジュール定義情報３がユーザ

定義インデクス４１を実現するモジュールの定義情報であることを示している。そして、関連するユーザ定義インデクスタイプの名称が「インデクスタイプ名」にＤＯＣ－ＩＮＤＥＸと示されている。その後に個々のモジュールに関する情報が指示される。

【００４９】

組込みモジュール"＿ｐ＿ｄｏｃ＿ｉｎｓｅｒｔ"は、起動契機“ＡＳ＿ＩＮＤＥＸ＿ＩＮＳＥＲＴ”によりテーブルデータ挿入時のインデクスメンテナンスの契機で起動されるモジュールであることを示している。

【００５０】

同様に、組込みモジュール"＿ｐ＿ｄｏｃ＿ｄｅｌｅｔｅ"は、起動契機“ＡＳ＿ＩＮＤＥＸ＿ＤＥＬＥＴＥ”によりテーブルデータ削除時のインデクスメンテナンスの契機で起動されるモジュールであることを、組込みモジュール"＿ｐ＿ｄｏｃ＿ｕｐｄａｔｅ"は、起動契機“ＡＳ＿ＩＮＤＥＸ＿ＵＰＤＡＴＥ”によりテーブルデータ更新時のインデクスメンテナンスの契機で起動されるモジュールであることを示している。

【００５１】

また、組込みモジュール”＿ｐ＿ｄｏｃ＿ｓｃａｎ”は、起動契機“ＡＳ＿ＩＮＤＥＸ＿ＳＣＡＮ”により本インデクスタイプのインデクスを用いた検索指示が与えられた際に、その検索を実現する為に起動されるモジュールであることを示している。

【００５２】

以上のインデクスタイプ定義（手順３０１）とインデクス実現モジュール定義情報登録（手順３０２）によって、インデクスを実現する為のモジュールをデータベース管理システム１が認知し、そのインデクスタイプのインデクスを作成及び使用することが可能になる。

【００５３】

次に、定義したインデクスタイプのインデクスを使用する為のインタフェースであるＡＤＴ関数を含むデータタイプの定義を行う（手順３０３）。データタイプの定義は、ＳＱＬの「ＣＲＥＡＴＥ　ＴＹＰＥ文」により行う。

【００５４】

図３の「ＣＲＥＡＴＥ　ＴＹＰＥ文」の例では、ｔｅｘｔ＿ｎｏ、ｔｅｘｔ＿ｎａｍｅ、ａｕｔｈｏｒ、ｃｏｎｔｅｎｔｓというそれぞれＩＮＴ、ＣＨＡＲ、ＣＨＡＲ、ＶＡＲＣＨＡＲ型データの属性と、ＢＯＯＬＥＡＮ型を返すＡＤＴ関数ＣＯＮＴＡＩＮＳを有する「ＴＥＸＴ」型ＡＤＴが定義されている。

【００５５】

データタイプ定義の手順３０３において定義したＡＤＴに関して、手順３０４において前記ＡＤＴを実現するモジュールのモジュール定義情報５４の登録を行う。以下に、図３のデータタイプ定義に基づいたＡＤＴ実現モジュールの定義情報の登録の際のユーザインタフェースの一例を挙げる。

【００５６】

```
adtmodule {
 ADT名:TEXT
 _p_text_contains{
    起動契機     :AS_FUNCTION、
    ADT関数名  :CONTAINS
 }
 _p_text_insert{
    起動契機     :AS_INSERT_TRIGGER
    ADT関数名  :NULL
 }
 _p_text_delete{
    起動契機     :AS_DELETE_TRIGGER
    ADT関数名  :NULL
 }
 _p_key_create{
    起動契機     :AS_KEY_CREATION
    ADT関数名  :NULL
    インデクスタイプ名称:DOC－INDEX
 }
}
```

まず、ａｄｔｍｏｄｕｌｅは、入力されたモジュール定義情報３がＡＤＴ実現モジュールの定義情報であることを示している。そして、関連するＡＤＴの名称が「ＡＤＴ名」にＴＥＸＴと示されている。その後に個々のモジュールに関する情報が指示される。

【００５７】

組込みモジュール”＿ｐ＿ｔｅｘｔ＿ｃｏｎｔａｉｎｓ”は、ＡＤＴ関数"ＣＯＮＴＡＩＮＳ”を実現する為に起動されることを示している。ＡＤＴ関数実現の為のモジュールには起動契機として“ＡＳ＿ＦＵＮＣＴＩＯＮ”を指定する。

【００５８】

組込みモジュール”＿ｐ＿ｔｅｘｔ＿ｉｎｓｅｒｔ”は、ＴＥＸＴ型データの挿入時（“ＡＳ＿ＩＮＳＥＲＴ＿ＴＲＩＧＧＥＲ”は挿入時が起動契機であることを示す）に起動されることを示している。

【００５９】

また、組込みモジュール”＿ｐ＿ｔｅｘｔ＿ｄｅｌｅｔｅ”は、ＴＥＸＴ型データを含む行の削除時（“ＡＳ＿ＤＥＬＥＴＥ＿ＴＲＩＧＧＥＲ”は削除時が起動契機であることを示す）に起動されることを示している。

【００６０】

また、“ＡＳ＿ＫＥＹ＿ＣＲＥＡＴＩＯＮ”で指示されている組込みモジュール"＿ｐ＿ｔｅｘｔ＿ｋｅｙｃｒｅａｔｅ"は、インデクスタイプ名称で指示されるＤＯＣ－ＩＮＤＥＸタイプのインデクスをメンテナンスする際のインデクスキー作成の為に起動されるモジュールであることを示している。

【００６１】

以上により、そのインデクスタイプのインデクスを使用及び維持（更新）する為のインタフェースが確立され、ユーザ問合せからのインデクス使用の取り決めが完了する。

【００６２】

上記インデクスタイプ及びＡＤＴの定義の下に、以下の手順３０５、手順３０６でデータベース領域４に格納するテーブルを定義し、そのテーブルに対して先に定義したユーザ定義インデクス４１を作成する。

【００６３】

テーブルの定義は、図３の手順３０５の例の様にＳＱＬにおいて「ＣＲＥＡＴＥ　ＴＡＢＬＥ文」により行う。この例では、ｔｉｔｌｅ、ｃｏｕｎｔｒｙ、ｐｒｏｄｕｃｅ＿ｙｅａｒ、ｇｕｉｄｅ、ｍｏｖｉｅ＿ｃｏｎｔｅｎｔｓというそれぞれＣＨＡＲ、ＩＮＴ、ＤＡＴＥ、ＴＥＸＴ、ＢＬＯＢ型の列から構成されるｍｏｖｉｅｓ＿ｌｉｂテーブルが定義されている。

【００６４】

ユーザ定義インデクス４１の作成を行うインデクス定義（手順３０６）において、作成しようとするインデクスに関する情報を登録する。インデクス定義では、作成するインデクスを識別する為のインデクス名称の他に次の３つの項目を指定する。

【００６５】

（Ａ）作成するインデクスのインデクスタイプ名称

（Ｂ）インデクスメンテナンス用入力キーを生成する為のデータを確定するテーブル名称、属性名、或いはＡＤＴ関数名

（Ｃ）定義したインデクスを使用する為のインタフェースとなるＡＤＴ関数名称

前記インデクス定義の際のインタフェースとして、図３の例で示してある「ＣＲＥＡＴＥ　ＩＮＤＥＸ文」の様なＳＱＬ形式に近い記述形式が考えられる。以下に、「ＣＲＥＡＴＥ　ＩＮＤＥＸ文」の構文記述例を変形ＢＮＦ記法で示す。

【００６６】

```
CREATE INDEX ［認可識別子.］ テーブル名称
    ON ｛列名
    |列名. . 属性名 ［、 列名. . 属性名 … ］
    |列名 （ADT関数名 ＩＮ ADT型名） ｝
  ＦＯＲ 関数名 ［、関数名］
  ＴＹＰＥ インデクスタイプ名称
```

ＴＹＰＥ句によって（Ａ）のどのタイプのインデクスを作成するかを指定する。ＯＮ句によって、（Ｂ）の指定テーブル名称のどの列、どの属性をインデクスメンテナンスの為のデータに使用するかを指定する。そして、ＦＯＲ句によって（Ｃ）のユーザ定義インデクス４１を使用する為のインタフェースとなるＡＤＴ関数を指定する。（Ｂ）の情報は、複数のテーブルデータ指定が可能である。

【００６７】

図３の「ＣＲＥＡＴＥ　ＩＮＤＥＸ文」では、作成インデクスタイプがＤＯＣ－ＩＮＤＥＸでありインデクス名称がＤＯＣ－ＳＥＡＲＣＨで、メンテナンスの為のテーブルデータ４２がｍｏｖｉｅｓ＿ｌｉｂテーブルのｇｕｉｄｅ列のｃｏｎｔｅｎｔｓ属性であるというインデクス定義がなされている。

【００６８】

また、「ＣＲＥＡＴＥ　ＩＮＤＥＸ文」においては、ディクショナリ５のインデクスタイプ定義情報５１を用いて、インデクス作成指示されているＯＮ句以下で指定されているテーブル列のデータ型に対して、指定インデクスタイプのインデクスが正当に作成できるかどうかのチェックを行うことも可能である。

【００６９】

以上の作成手順によりインデクスタイプ定義及びインデクス定義（作成）が行われるが、あるインデクスタイプを適用するインデクスの構成インデクスキーが、あるＡＤＴデータを指定する場合、すなわちあるインデクスタイプとデータタイプを同時にデータベース管理システム１に組込む場合、図３の手順例の様な作成手順ではなく、（３）データタイプ定義及び（４）ＡＤＴ実現モジュール定義情報登録を、（１）インデクスタイプ定義及び（２）インデクス実現モジュール定義情報登録の前に行うことも可能である。

【００７０】

本実施形態のデータベース処理方法を実施する実施装置において、先に示した各種定義情報を格納するディクショナリ５の一構成例について図４、図５及び図６を用いて説明する。

【００７１】

図４は、本実施形態のデータベース処理方法を実施する実施装置のディクショナリ５に格納されるインデクスタイプ定義情報５１の一構成例を示す図である。図４において、５１ａはインデクスタイプ情報、５１１はインデクスタイプ情報エントリ、５１ｂはインデクスキー情報、５１２はインデクスキー情報エントリである。

【００７２】

図４に示す様に、本実施形態のデータベース処理方法を実施する実施装置のディクショナリ５に格納されるインデクスタイプ定義情報５１では、インデクスタイプ情報５１ａとインデクスキー情報５１ｂがある。

【００７３】

インデクスタイプ情報５１ａは、ユーザ定義インデクスタイプ数分のインデクスタイプ情報エントリ５１１から成り、インデクスタイプ情報エントリ５１１は、インデクスタイプ名称、インデクスタイプｉｄ、インデクスキー数を含んでいる。

【００７４】

インデクスタイプ名称は、インデクスタイプ定義によりユーザから指定されたインデクスタイプの名称を示しており、インデクスタイプｉｄはインデクスタイプ定義時インデクスタイプ名称に対してデータベース管理システム１が割り当てた識別子である。インデクスタイプｉｄは、データベース管理システム１内にてインデクスタイプ名称の替わりにインデクスタイプの識別子として使用される。

【００７５】

インデクスキー数は、そのインデクスタイプのインデクスをメンテナンスする為に必要なインデクスキーの数であり、インデクスキー数は、「ＣＲＥＡＴＥ　ＩＮＤＥＸ　ＴＹＰＥ文」により指定されたインデクスキーデータ型の数を示している。

【００７６】

インデクスキー情報５１ｂは、各インデクスタイプのインデクスキー数の総和数分のインデクスキー情報エントリ５１２から成る。

【００７７】

インデクスキー情報エントリ５１２は、どのインデクスタイプに関するインデクスキー情報５１ｂかを示すインデクスタイプｉｄ、そのインデクスキーのデータ型、インデクスキーの定義長、またそのインデクスキーの指定番号を含む。ここで前記指定番号は、「ＣＲＥＡＴＥ　ＩＮＤＥＸ　ＴＹＰＥ文」にて指定された順序を示している。

【００７８】

インデクスタイプ情報エントリ５１１とインデクスキー情報エントリ５１２はインデクスタイプｉｄにより１対ｎ（ｎはインデクスキー数）に関連付けられている。

【００７９】

図５は、本実施形態のデータベース処理方法を実施する実施装置のディクショナリ５に格納されるインデクス定義情報５２の一構成例を示す図である。図５において、５２ａはインデクス定義情報、５２１はインデクス定義情報エントリ、５２ｂはインデクスキー生成用データ情報、５２２はインデクスキー生成用データ情報エントリ、５２ｃはインデクスアクセス用ＡＤＴ関数情報、５２３はインデクスアクセス用ＡＤＴ関数情報エントリであ

る。
【００８０】
図５に示す様に、本実施形態のデータベース処理方法を実施する実施装置のディクショナリ５に格納されるインデクス定義情報５２では、インデクス定義情報５２ａ、インデクスキー生成用データ情報５２ｂ、そしてインデクスアクセス用ＡＤＴ関数情報５２ｃの３つがある。
【００８１】
インデクス定義情報５２ａは、作成インデクス数分のインデクス定義情報エントリ５２１から成り、インデクス定義情報エントリ５２１は、インデクス名称、インデクスｉｄ、キー生成用データ数、インデクスタイプｉｄを含んでいる。
【００８２】
インデクス名称は、インデクス定義によりユーザから指定されたインデクスの名称であり、インデクスｉｄはインデクス定義時インデクス名称に対してデータベース管理システム１が割り当てた識別子である。インデクスｉｄは、データベース管理システム１内にてインデクス名称の替わりにインデクスの識別子として使用される。
【００８３】
キー生成用データ数とは、「ＣＲＥＡＴＥ　ＩＮＤＥＸ文」により指定されたインデクスをメンテナンスする為に必要なテーブルデータ４２の項目数である。そしてインデクスタイプｉｄがそのインデクスのタイプを示している。
【００８４】
インデクスキー生成用データ情報５２ｂは、インデクスキーを生成する為のテーブルデータ４２に関する情報であり、この情報に従って必要なテーブルデータ４２を取得する。インデクスキー生成用データ情報５２ｂは、各インデクスのキー生成用データ数の総和数分のインデクスキー生成用データ情報エントリ５２２から成る。
【００８５】
インデクスキー生成用データ情報エントリ５２２は、どのインデクスに関するインデクスキー生成用データ情報５２ｂかを示すインデクスｉｄ、指定番号、テーブル名称、列名称、属性名または関数名を含む。ここで指定番号は、「ＣＲＥＡＴＥ　ＩＮＤＥＸ文」にて指定された順序を示している。
【００８６】
インデクスアクセス用ＡＤＴ関数情報５２ｃは、インデクスを使用する為のインタフェースとなるＡＤＴ関数に関する情報であり、作成インデクス数分のインデクスアクセス用ＡＤＴ関数情報エントリ５２３からなる。
【００８７】
インデクスアクセス用ＡＤＴ関数情報エントリ５２３は、どのインデクスに関するインデクスアクセス用ＡＤＴ関数情報５２ｃかを示すインデクスｉｄと、インタフェースとなるＡＤＴ関数名を含む。
【００８８】
ここで、インデクスアクセス用ＡＤＴ関数情報エントリ５２３は、どのインデクスに関するインデクスアクセス用ＡＤＴ関数情報５２ｃかを示すインデクスｉｄ、またインタフェースとなるＡＤＴ関数名を含む。
【００８９】
ここで、複数の異なるＡＤＴ関数名と特定のインデクスを示すインデクスｉｄとを対応付けることにより、特定のユーザ定義インデクス４１や各種モジュール１００ａ～１００ｄの特定の実現モジュールを複数の異なるＡＤＴ定義関数で共有することが可能である。
【００９０】
図６は、本実施形態のデータベース処理方法を実施する実施装置のディクショナリ５に格納されるモジュール定義情報５４の一構成例を示す図である。図６において、５４ａはインデクスモジュール定義情報、５４１はインデクスモジュール定義情報エントリ、５４ｂはＡＤＴモジュール定義情報、５４２はＡＤＴモジュール定義情報エントリである。

【００９１】
図６に示す様に、本実施形態のデータベース処理方法を実施する実施装置のディクショナリ５に格納されるモジュール定義情報５４では、インデクスモジュール定義情報５４ａとＡＤＴモジュール定義情報５４ｂの２つがある。それらは各々、ユーザ定義インデクス４１を実現する為のモジュールに関する定義情報、ユーザ定義データタイプ（ＡＤＴ）を実現する為のモジュールに関する定義情報である。
【００９２】
インデクスモジュール定義情報５４ａは、ユーザ定義インデクス４１を実現する為のモジュール数分のインデクスモジュール定義情報エントリ５４１から成り、インデクスモジュール定義情報エントリ５４１は、ユーザ定義インデクス４１のアクセスを実現する指定モジュールに関連するモジュール名、インデクスタイプｉｄ、起動契機を含んでいる。
【００９３】
インデクスタイプｉｄは、どのインデクスタイプを実現するかを示すインデクスｉｄである。起動契機は、関連モジュールがユーザ定義インデクス４１を実現する為に起動される契機を示している。
【００９４】
ＡＤＴモジュール定義情報５４ｂは、ＡＤＴ実現の為のモジュール数分のＡＤＴモジュール定義情報エントリ５４２から成り、ＡＤＴモジュール定義情報エントリ５４２は、ＡＤＴの振る舞いを実現する指定モジュールに関連するモジュール名、ＡＤＴ名、起動契機、起動ＡＤＴ関数名、インデクスタイプｉｄを含んでいる。
【００９５】
ＡＤＴ名は、その関連するモジュール（モジュール名により識別される）が、どのＡＤＴの振る舞いを実現する為に起動されるか、またはどのＡＤＴに対する操作を行った際に起動されるかを示している。
【００９６】
起動契機は、関連モジュールがユーザ定義インデクス４１を実現する為に起動される契機を示しており、もし、その関連するモジュールがＡＤＴの振る舞いを実現する為のもの、すなわちユーザからの問合せ要求（ＳＱＬ文）内に明示的に記述されるＡＤＴ関数を実現する為のものである場合、起動契機にはユーザの指定に伴い”ＡＳ＿ＦＵＮＣＴＩＯＮ”が設定される。
【００９７】
起動契機に”ＡＳ＿ＦＵＮＣＴＩＯＮ”が設定されている場合、そのモジュールを起動するＡＤＴ関数の名称が起動ＡＤＴ関数名に示される。
【００９８】
ユーザからのモジュール定義情報３にモジュールの起動契機として”ＡＳ＿ＫＥＹ＿ＣＲＥＡＴＩＯＮ”が指定された場合、ＡＤＴモジュール定義情報５４ｂの起動契機に”ＡＳ＿ＫＥＹ＿ＣＲＥＡＴＩＯＮ”が設定されると共に、インデクスタイプｉｄには、どのインデクスタイプのインデクスの為のインデクスキー生成モジュールかを示すインデクスタイプｉｄが設定される。
【００９９】
また、インデクスタイプ定義情報５１及びインデクス定義情報５２のインデクスタイプ名称及びインデクスｉｄと、モジュール定義情報５４のモジュール名及びインデクスｉｄとを変更することにより、特定のＡＤＴ関数名に対して起動されるモジュールを変更して、データベース処理の内容を変更することが可能。
【０１００】
次に図３で示した新たなインデクスを追加作成する際の各手順毎の処理を図７から図９のフローチャートを用いて詳細に説明する。
【０１０１】
図７は、本実施形態のデータベース処理方法を実施する実施装置のインデクスタイプ定義処理の処理手順を示すフローチャートである。

【０１０２】
図７に示す様に、本実施形態のデータベース処理方法を実施する実施装置のインデクスタイプ定義処理では、定義処理部８０におけるインデクスタイプ定義手順３０１の処理の流れの一例を表しており、まず、ステップ８１１において、ユーザから入力されたインデクスタイプ名称に対してインデクスタイプを識別する為のインデクスタイプｉｄを決定して割り当てる。
【０１０３】
そして、そのインデクスタイプｉｄ及びユーザ入力情報からインデクスタイプ情報エントリ５１１を作成し、ディクショナリ５への登録を行う（ステップ８１２）。ディクショナリ５への登録は、定義処理部８０からの要求を受けてディクショナリ管理部９０が行う。
【０１０４】
次に同様にして、インデクスタイプｉｄを用いてインデクスキー情報エントリ５１２を作成しディクショナリ５へ登録する（ステップ８１３）。
【０１０５】
図３の例では、ユーザが指定したインデクスタイプＤＯＣ－ＩＮＤＥＸに対して、例えばインデクスタイプｉｄ「８９０」を割り当てる。また、インデクスキーのデータ型はＶＡＲＣＨＡＲ一つであると指定されているので、まず、以下の様なインデクスタイプ情報エントリ５１１を作成し、図４に示した様にディクショナリ５に登録する。
【０１０６】
インデクス名称　　：ＤＯＣ－ＩＮＤＥＸ
インデクスタイプｉｄ：８９０
インデクスキー数　：１
インデクスキー情報５１ｂに関しては、ＶＡＲＣＨＡＲデータの指定長が３２０００であり、インデクスキー指定が一つであることから指定番号も１である様な以下の一インデクスキー情報エントリ５１２を作成し、図４に示した様にディクショナリ５に登録する。
【０１０７】
インデクスタイプｉｄ：８９０
データ型　　：ＶＡＲＣＨＡＲ
定義長　　：３２０００
指定番号　　：１
図８は、本実施形態のデータベース処理方法を実施する実施装置のモジュール定義情報登録処理の処理手順を示すフローチャートである。
【０１０８】
図８に示す様に、本実施形態のデータベース処理方法を実施する実施装置のモジュール定義情報登録処理では、図３で説明したインデクス実現モジュール定義情報登録の手順３０２及びＡＤＴ実現モジュール定義情報登録の手順３０４を実行するモジュール定義情報登録部７０における処理の流れを表している。
【０１０９】
まず、ステップ７１において、ユーザ定義インデクス４１を追加するユーザから入力されたモジュール定義情報３が、インデクスモジュールの定義情報であるかＡＤＴモジュールの定義情報であるかを判定する。
【０１１０】
ここで、入力されたモジュール定義情報３がインデクスモジュールの定義情報である場合にはステップ７２に進み、ディクショナリ５に格納されているインデクスタイプ情報５１ａを参照し、ユーザがモジュール定義情報３により指定したインデクスタイプ名称からインデクスタイプｉｄを取得する。
【０１１１】
そして、そのインデクスタイプｉｄとユーザ入力情報からインデクスモジュール定義情報エントリ５４１を作成し、ディクショナリ５への登録を行い（ステップ７３）、処理を終了する（ステップ７８）。

【０１１２】
ステップ７１の入力されたモジュール定義情報３がＡＤＴモジュールの定義情報である場合には、ステップ７４の起動契機判定処理に進み、ユーザからのモジュール契機指定が、“ＡＳ＿ＦＵＮＣＴＩＯＮ”指定か、“ＡＳ＿ＫＥＹ＿ＣＲＥＡＴＩＯＮ”指定か、それ以外かを判定する。
【０１１３】
ユーザからのモジュール契機指定が“ＡＳ＿ＦＵＮＣＴＩＯＮ”指定の場合には、そのモジュールによって実現されるＡＤＴ関数の名称をモジュール定義情報エントリ作成の為に保持し（ステップ７６）、ステップ７７に進む。
【０１１４】
ユーザからのモジュール契機指定が“ＡＳ＿ＫＥＹ＿ＣＲＥＡＴＩＯＮ”指定の場合には、ディクショナリ５に格納されているインデクスタイプ情報５１ａを参照し、ユーザがモジュール定義情報３により指定したインデクスタイプ名称からインデクスタイプｉｄを取得保持し（ステップ７５）、ステップ７７に進む。
【０１１５】
ユーザからのモジュール契機指定が“ＡＳ＿ＦＵＮＣＴＩＯＮ”指定や“ＡＳ＿ＫＥＹ＿ＣＲＥＡＴＩＯＮ”指定以外の場合には、そのままステップ７７に進む。
【０１１６】
ステップ７７では、ステップ７５またはステップ７６において保持しておいた情報を元にＡＤＴモジュール定義情報エントリ５４２を作成し、それをディクショナリ５へ登録して処理を終了する（ステップ７８）。
【０１１７】
起動契機が“ＡＳ＿ＦＵＮＣＴＩＯＮ”のエントリには、起動ＡＤＴ関数名に保持してあったＡＤＴ関数名が設定される。また、起動契機が“ＡＳ＿ＫＥＹ＿ＣＲＥＡＴＩＯＮ”のエントリには、インデクスタイプｉｄに保持してあったインデクスタイプｉｄが設定される。ディクショナリ５への登録は、ディクショナリ管理部９０が代行して行う。
【０１１８】
図９は、本実施形態のデータベース処理方法を実施する実施装置のインデクス定義処理の処理手順を示すフローチャートである。
【０１１９】
図９に示す様に、本実施形態のデータベース処理方法を実施する実施装置のインデクス定義処理では、定義処理部８０におけるインデクス定義（手順３０６）の処理の流れの一例を表しており、まず、ステップ８２１において、ユーザから入力されたインデクス名称に対してインデクスを識別する為のインデクスｉｄを決定して割り当てる。
【０１２０】
そして、そのインデクスｉｄ及びユーザ入力情報からインデクス定義情報エントリ５２１を作成し、ディクショナリ５への登録を行う（ステップ８２２）。ディクショナリ５への登録は、定義処理部８０からの要求を受けてディクショナリ管理部９０か行う。
【０１２１】
同様にして、インデクスｉｄ及びインデクスを作成する為に必要なテーブルデータ４２をアクセスする為の情報（ユーザ入力情報）からインデクスキー生成用データ情報エントリ５２２を作成し、ディクショナリ５へ登録する（ステップ８２３）。
【０１２２】
インデクスキー生成用データ情報エントリ５２２は、インデクス定義情報エントリ５２１内のキー生成用データ数分作成される。そして、エントリ内の指定番号には「ＣＲＥＡＴＥ　ＩＮＤＥＸ文」で記述されたインデクスキー生成用データの順序番号が設定される。
【０１２３】
次に、インデクスｉｄ及びインデクスを使用する為のインタフェースである指定ＡＤＴ関数名を用いて、インデクスアクセス用ＡＤＴ関数情報エントリ５２３を作成してディクショナリ５へ登録し（ステップ８２４）、処理を終了する（ステップ８２５）。

【０１２４】
インデクスアクセス用ＡＤＴ関数情報エントリ５２３も、「ＣＲＥＡＴＥ　ＩＮＤＥＸ文」で指定された数分作成される。尚、図３に示した例ではインデクスＤＯＣ－ＩＮＤＥＸに関するエントリは、ＡＤＴ関数名”ＣＯＮＴＡＩＮＳ”を含むエントリ一つだけである。
【０１２５】
図１０は、本実施形態のデータベース処理方法を実施する実施装置のＡＤＴ関数解析処理の処理手順を示すフローチャートである。
【０１２６】
図１０に示す様に、本実施形態のデータベース処理方法を実施する実施装置のＡＤＴ関数解析処理では、問合せ解析処理部１０におけるユーザ要求２（ＳＱＬ文）内にＡＤＴ関数が出てきた際のＡＤＴ関数に関する解析処理の流れの一例を表している。
【０１２７】
まず、ステップ１２において、解析対象のＡＤＴ関数名、そのＡＤＴ関数を適用するテーブル名称及び列名称を用いて、ディクショナリ５に格納されているインデクスアクセス用ＡＤＴ関数情報エントリ５２３、インデクス定義情報エントリ５２１及び関連インデクスキー生成用データ情報エントリ５２２を取得する。
【０１２８】
次に、ディクショナリ５内に関連するエントリが存在するか、すなわち取得エントリがあったかを判定する（ステップ１３）。関連するインデクス定義情報５２が存在する場合、取得情報のインデクスｉｄで識別されるインデクスを用いて要求ＡＤＴ関数を実現するアクセス経路を用いることを決定する（ステップ１４）。
【０１２９】
次にステップ１５にて、使用を決定したインデクスをアクセスする為の情報として、インデクス検索用モジュール起動情報を、ディクショナリ５内のインデクスモジュール定義情報５４ａからインデクスタイプｉｄ（インデクス定義情報エントリ５２１内）を用いて取得する。
【０１３０】
そして、ステップ１８にて、使用を決定したインデクスのインデクスｉｄ及びそのインデクスのアクセスを実現する為のモジュール情報（モジュール名）を用いて、インデクスアクセス用の内部処理コードを生成し、処理を終了する（ステップ１９）。
【０１３１】
ステップ１３の判定処理において、該当するインデクス定義情報５２がない場合、インデクスは使用せず、一行データに対してＡＤＴ関数実現モジュールを起動することによりＡＤＴ関数を実現するというアクセス経路に決定する（ステップ１６）。
【０１３２】
そして、ステップ１７にて、関連ＡＤＴ関数名を用いてＡＤＴ関数実現モジュール情報を取得し、その情報（モジュール名）を用いてデータベース処理用内部処理コードを生成する（ステップ１８）。
【０１３３】
生成されたデータベース処理用内部処理コードの指示に従って、データベース演算処理部２０において、ユーザ定義インデクス管理部３０及びテーブルデータ管理部４０が連携してデータベース処理を実行する。
【０１３４】
図１１は、本実施形態のデータベース処理方法を実施する実施装置のインデクス処理の処理手順を示すフローチャートである。
【０１３５】
図１１に示す様に、本実施形態のデータベース処理方法を実施する実施装置のインデクス処理では、ユーザ定義インデクス管理部３０における処理を表しており、データベース演算処理部２０からの要求を受け取った際のユーザ定義インデクス管理部３０におけるイン

デクス処理の一例を示している。

【０１３６】

まず、ステップ３０１では、処理要求がインデクス検索要求であるかまたは更新要求であるかを判定する。処理要求がインデクス検索要求の場合、ステップ３１０に進み、インデクス検索を実現するモジュールを起動する為の情報を取得する。そして、その情報を基にインデクス検索モジュールを起動し、インデクス検索を行う（ステップ３１１）。インデクスを検索した結果をデータベース処理部に返し（ステップ３１２）、処理を終了する（ステップ３０９）。

【０１３７】

ステップ３０１の判定結果が更新処理である場合、インデクスの更新処理の前準備としてインデクスを更新する為のデータの準備を行う。このインデクスの更新処理に対する入力データにはＡＤＴモジュール定義情報５４ｂにてユーザが指定したキー生成モジュールを用いて生成するものと、テーブルデータ４２を参照して取得したデータをそのまま用いるものとの二種類がある。そして、ステップ３０２において、キー生成モジュール起動するモジュール定義情報５４があるかどうかでその判定を行う。

【０１３８】

関連するキー生成モジュール起動情報がある場合、その情報を取得し（ステップ３０３）、取得情報を用いてキー生成モジュールを起動してキーの生成を行う（ステップ３０４）。そして、生成したキーをインデクス更新用入力キーに指定する（ステップ３０５）。

【０１３９】

また、関連するキー生成モジュール起動情報が無い場合、更新行のデータをインデクス更新用キーに指定する（ステップ３０６）。

【０１４０】

次にステップ３０７において、インデクス更新モジュールを起動する為の情報を取得する。そして、先ほどの指定インデクス更新用キーを用いて、インデクス更新モジュールを起動することによりインデクスの更新を行う（ステップ３０８）。

【０１４１】

次に、図１で説明した一構成及び図３のデータベースの定義（ユーザ定義インデクス４１組み込み）の下で、具体的なコマンド例を用いてユーザ組込みインデクスを用いたデータベース処理について説明する。

【０１４２】

まず、以下のユーザからの更新要求（ＳＱＬ文）を例に説明する。

【０１４３】

ＩＮＳＥＲＴ ＩＮＴＯ ｍｏｖｉｅｓ＿ｌｉｂ
ＶＡＬＵＥＳ（”ｉｎｄｅｐｅｎｄｅｎｃｅ ｄａｙ”、１２、１９９６、ＴＥＸＴ（８７３４、”ＩＤ４”、”Ｍｒ．Ｘ”、ｔｅｘｔ＿ｖ）、ｃｏｎｔｅｎｔｓ＿ｖ）

ここで、ＡＤＴ関数ＴＥＸＴはＡＤＴデータ（インスタンス）を生成するコンストラクタ関数であり、テキスト番号（８７４３）、テキスト名称（"ＩＤ４"）、著者（"Ｍｒ．Ｘ"）、テキストの中身（変数ｔｅｘｔ＿ｖで示されるＶＡＲＣＨＡＲ型データ）を引数としＴＥＸＴ型データを生成する。

【０１４４】

上記ＳＱＬ文の例では、ｔｉｔｌｅ（題名）：”ｉｎｄｅｐｅｎｄｅｎｃｅ ｄａｙ”、ｃｏｕｎｔｒｙ（制作国名コード）：１２、ｐｒｏｄｕｃｅ＿ｙｅａｒ（制作年）：１９９６、ｇｕｉｄｅ（解説）：ＴＥＸＴ型データ（コンストラクタ関数で生成される）、ｍｏｖｉｅ＿ｃｏｎｔｅｎｔｓ（映像）：変数ｃｏｎｔｅｎｔｓ＿ｖで示されるＢＬＯＢ型データ、を値に持つデータのｍｏｖｉｅｓ＿ｌｉｂテーブルへの挿入要求を表している。また、ｃｏｕｎｔｒｙ列に指定されているコード１２は制作国ＵＳＡを示すコードである。

【０１４５】

図１１のフローチャートに従ってユーザ定義インデクス管理部３０でテーブルデータ４２

の挿入に伴いユーザ定義メンテナンスを行う。

【０１４６】

図５に示す様に、ｍｏｖｉｅｓ＿ｌｉｂテーブルのインデクス定義情報５２ａには、インデクスＤＯＣ－ＳＥＡＲＣＨが作成されているので、インデクスＤＯＣ－ＳＥＡＲＣＨの更新処理を行う。

【０１４７】

まず、インデクスキー生成用データ情報５２ｂに基づきｇｕｉｄｅ列のｃｏｎｔｅｎｔｓ属性データをインデクスキー生成用データとして準備する。

【０１４８】

そして、準備したデータを入力としてＡＤＴモジュール定義情報５４ｂで示される＿ｐ＿ｔｅｘｔ＿ｋｅｙｃｒｅａｔｅモジュールを起動し、ＤＯＣ－ＳＥＡＲＣＨインデクス更新の為のインデクスキーを作成する。生成されるインデクスキーは定義長３２０００のＶＡＲＣＨＡＲ型データである。

【０１４９】

次に作成したインデクスキーを入力としてインデクスモジュール定義情報ＡＳ＿ＩＮＤＥＸ＿ＩＮＳＥＲＴ契機で示されるモジュール＿ｐ＿ｄｏｃ＿ｉｎｓｅｒｔを起動することによりインデクス更新処理を行う。

【０１５０】

次に、以下のユーザからの検索要求（ＳＱＬ文）を例に説明する。

【０１５１】

ＳＥＬＥＣＴ ｔｉｔｌｅ、ｍｏｖｉｅ＿ｃｏｎｔｅｎｔｓ ＦＲＯＭ ｍｏｖｉｅｓ＿ｌｉｂ ＷＨＥＲＥ ＣＯＮＴＡＩＮＳ（ｇｕｉｄｅ、"ｉｎｄｅｐｅｎｄｅｎｃｅ”）

ここで、関数ＣＯＮＴＡＩＮＳは、第一パラメータである指定ＡＤＴデータ内に、第二パラメータにより指定される単語を含む場合にＴＲＵＥ（ＢＯＯＬＥＡＮ型）を返す関数である。

【０１５２】

上記ＳＱＬ文の例では、ｍｏｖｉｅｓ＿ｌｉｂテーブルのｇｕｉｄｅ列（解説）に”ｉｎｄｅｐｅｎｄｅｎｃｅ”を含むテーブルデータ４２のｔｉｔｌｅ（題名）及びｍｏｖｉｅ＿ｃｏｎｔｅｎｔｓ（映像）を取り出す検索要求である。

【０１５３】

問合せ解析処理部１０において、図１０のフローチャートに従って、ＡＤＴ関数の実現はＤＯＣ－ＩＮＤＥＸユーザ定義インデクスタイプであるインデクスＤＯＣ－ＳＥＡＲＣＨを用いて行うことに決定される。

【０１５４】

そして、インデクス検索モジュール＿ｐ＿ｔｅｘｔ＿ｃｏｎｔａｉｎｓを起動することにより検索結果としてテーブルデータ４２が確定され、テーブルデータ管理部４０により指定取り出し列であるｔｉｔｌｅ及びｍｏｖｉｅ＿ｃｏｎｔｅｎｔｓの値がユーザに問合わせ結果として返る。

【０１５５】

以上示したフローチャートの処理は、図２で例として示したコンピュータシステム１０００におけるプログラムとして実行される。

【０１５６】

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１０００に物理的に直接接続される外部記憶装置１００３のみに格納されるものではなく、コンピュータシステム１０００に物理的に直接接続されてい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されていても良い。

【０１５７】

以上説明した様に、本実施形態のデータベース処理方法を実施する実施装置によれば、処理要求中のユーザ定義関数がインデクス定義情報に定義されたものである場合に、前記ユーザ定義関数に対応する実現モジュールを実行してインデクスアクセスを行うので、特定

のユーザ定義関数に依存しない汎用的なデータベース処理を行うことが可能である。

【０１５８】

また、本実施形態のデータベース処理方法を実施する実施装置によれば、複数の異なるユーザ定義関数と特定のインデクスとをインデクス定義情報で対応付けることにより、データベース領域のデータに特定の処理を行う実現モジュールを複数の異なるユーザ定義関数で共有するので、特定のインデクスアクセスを行う実現モジュールをユーザ定義関数から独立させることが可能である。

【０１５９】

また、本実施形態のデータベース処理方法を実施する実施装置によれば、インデクス定義情報及びモジュール定義情報を変更することにより、特定のユーザ定義関数に対応付けられた実現モジュールを変更するので、必要な実現モジュールのみを内蔵することが可能である。

【０１６０】

以上、本発明を前記実施形態に基づき具体的に説明したが、本発明は、前記実施形態に限定されるものではなく、その要旨を逸脱しない範囲において種々変更可能であることは勿論である。

【０１６１】

【発明の効果】

本願において開示される発明のうち代表的なものによって得られる効果を簡単に説明すれば、下記のとおりである。

【０１６２】

すなわち、処理要求中のユーザ定義関数がインデクス定義情報に定義されたものである場合に、前記ユーザ定義関数に対応する実現モジュールを実行してインデクスアクセスを行うので、特定のユーザ定義関数に依存しない汎用的なデータベース処理を行うことが可能である。

【図面の簡単な説明】

【図１】本実施形態のデータベース処理方法を実施する実施装置の概略構成を示す図である。

【図２】本実施形態のデータベース処理方法を実施する実施装置のハードウェア構成の一例を示す図である。

【図３】本実施形態のデータベース処理方法を実施する実施装置のインデクス追加作成手順の一例を示す図である。

【図４】本実施形態のデータベース処理方法を実施する実施装置のディクショナリ５に格納されるインデクスタイプ定義情報５１の一構成例を示す図である。

【図５】本実施形態のデータベース処理方法を実施する実施装置のディクショナリ５に格納されるインデクス定義情報５２の一構成例を示す図である。

【図６】本実施形態のデータベース処理方法を実施する実施装置のディクショナリ５に格納されるモジュール定義情報５４の一構成例を示す図である。

【図７】本実施形態のデータベース処理方法を実施する実施装置のインデクスタイプ定義処理の処理手順を示すフローチャートである。

【図８】本実施形態のデータベース処理方法を実施する実施装置のモジュール定義情報登録処理の処理手順を示すフローチャートである。

【図９】本実施形態のデータベース処理方法を実施する実施装置のインデクス定義処理の処理手順を示すフローチャートである。

【図１０】本実施形態のデータベース処理方法を実施する実施装置のＡＤＴ関数解析処理の処理手順を示すフローチャートである。

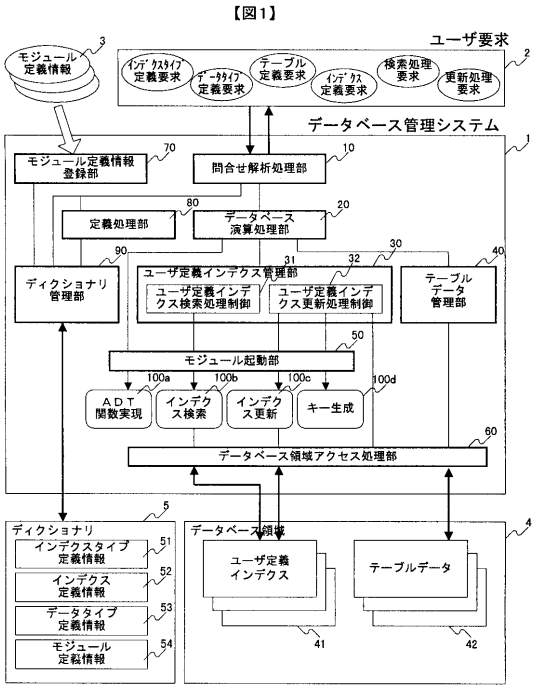
【図１１】本実施形態のデータベース処理方法を実施する実施装置のインデクス処理の処理手順を示すフローチャートである。

【符号の説明】

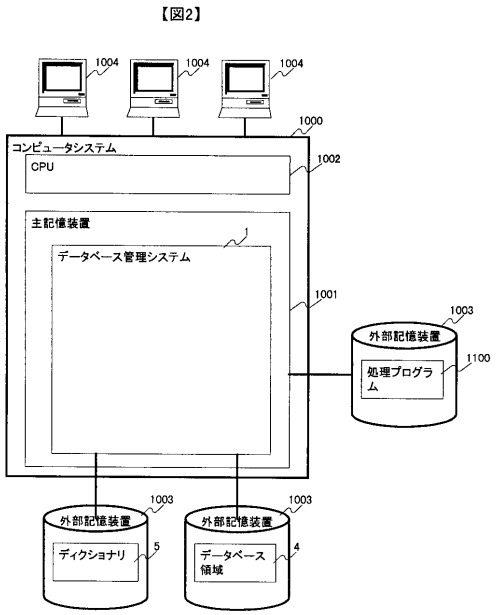
１…データベース管理システム、２…ユーザ要求、３…モジュール定義情報、４…データ

ベース領域、５…ディクショナリ、１０…問合せ解析処理部、２０…データベース演算処理部、３０…ユーザ定義インデクス管理部、３１…ユーザ定義インデクス検索処理制御部、３２…ユーザ定義インデクス更新処理制御部、４０…テーブルデータ管理部、４１…ユーザ定義インデクス、４２…テーブルデータ、５０…モジュール起動部、５１…インデクスタイプ定義情報、５２…インデクス定義情報、５３…データタイプ定義情報、５４…モジュール定義情報、６０…データベース領域アクセス処理部、７０…モジュール定義情報登録部、８０…定義処理部、９０…ディクショナリ管理部、１００ａ～１００ｄ…各種モジュール、１０００…コンピュータシステム、１００１…主記憶装置、１００２…ＣＰＵ、１００３…外部記憶装置、１００４…端末、１１００…処理プログラム、３０１～３０６…手順、５１ａ…インデクスタイプ情報、５１１…インデクスタイプ情報エントリ、５１ｂ…インデクスキー情報、５１２…インデクスキー情報エントリ、５２ａ…インデクス定義情報、５２１…インデクス定義情報エントリ、５２ｂ…インデクスキー生成用データ情報、５２２…インデクスキー生成用データ情報エントリ、５２ｃ…インデクスアクセス用ＡＤＴ関数情報、５２３…インデクスアクセス用ＡＤＴ関数情報エントリ、５４ａ…インデクスモジュール定義情報、５４１…インデクスモジュール定義情報エントリ、５４ｂ…ＡＤＴモジュール定義情報、５４２…ＡＤＴモジュール定義情報エントリ。

【図１】

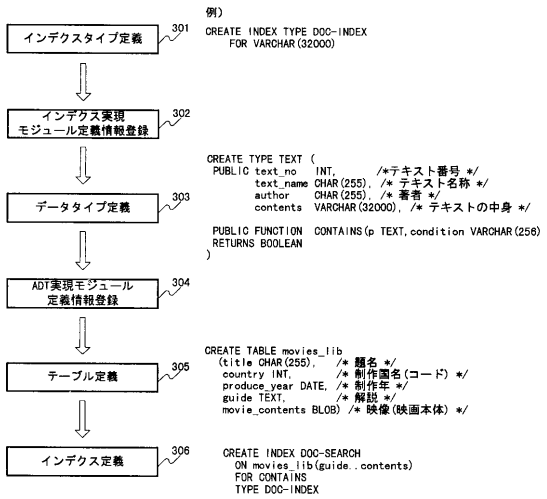


【図２】



【図３】

【図3】



【図４】

【図4】

インデクスタイプ情報 51a

インデクスタイプ名称	インデクスタイプid	インデクスキー数	
DOC-INDEX	890	1	511
XYZ-INDEX	235	2	511
…	…	…	511

インデクスキー情報 51b

インデクスタイプid	データ型	定義長	指定番号	
890	VARCHAR	32000	1	512
235	INT	4	1	512
235	CHAR	256	2	512
…	…	…	…	512

【図５】

【図5】

インデクス定義情報 52a

インデク名称	インデクスid	キー生成用データ数	インデクスタイプid	
DOC-SEARCH	5001	1	890	521
…	…	…	…	521

インデクスキー生成用データ情報 52b

インデクスid	指定番号	テーブル名称	列名称	属性名/関数名	
5001	1	movies_lib	guide	contents	522
…	…	…	…	…	522

インデクスアクセス用ADT関数情報 52c

インデクスid	ADT関数名	
5001	CONTAINS	523
…	…	523

【図６】

【図6】

インデクスモジュール定義情報 54a

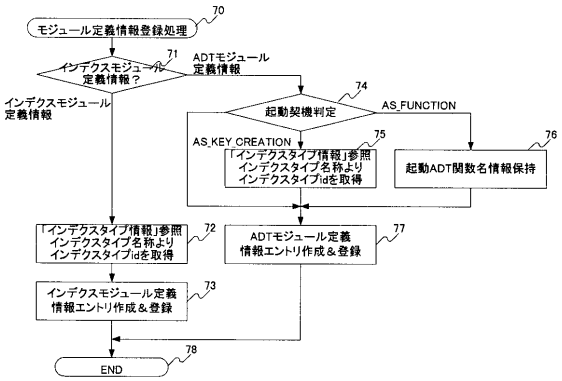
モジュール名	インデクスタイプid	起動契機	
_p_doc_insert	890	AS_INDEX_INSERT	541
_p_doc_delete	890	AS_INDEX_DELETE	541
_p_doc_update	890	AS_INDEX_UPDATE	541
_p_doc_scan	890	AS_INDEX_SCAN	541
…	…	…	541

ADTモジュール定義情報 54b

モジュール名	ADT名	起動契機	起動ADT関数名	インデクスタイプid	
_p_text_contains	TEXT	AS_FUNCTION	CONTAINS	–	542
_p_text_keycreate	TEXT	AS_KEY_CREATION	–	890	542
_p_text_insert	TEXT	AS_INSERT_TRIGGER	–	–	542
_p_text_delete	TEXT	AS_DELETE_TRIGGER	–	–	542
…	…	…	…	…	542

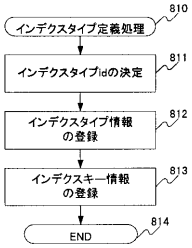
【図８】

【図8】



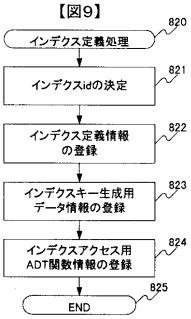
【図７】

【図7】



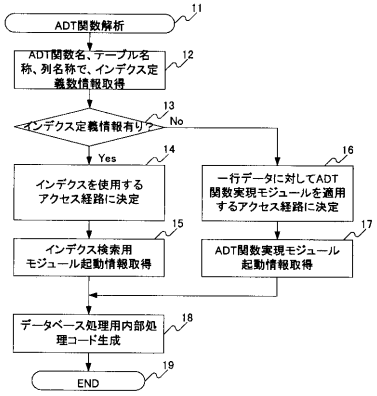
【図９】

【図9】



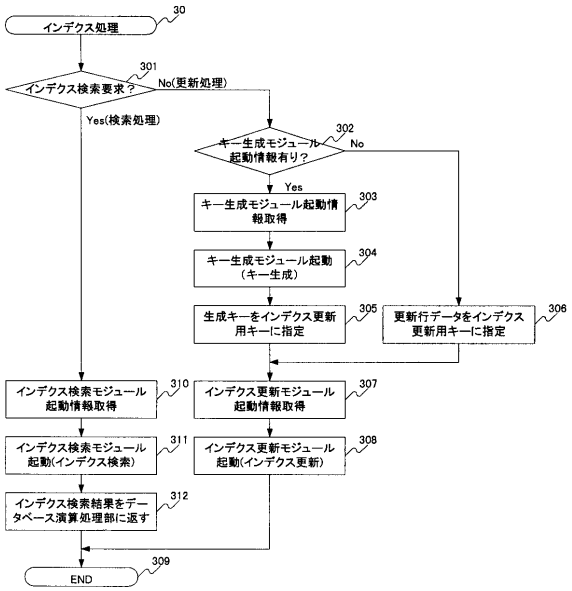
【図１０】

【図10】



【図１１】

【図11】



フロントページの続き

(72)発明者　亀城　嘉人
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　ソフトウェア開発本部内

審査官　相崎　裕恒

(56)参考文献　中村，ＯＲＤＢへの主役交代始まる　オブジェクト指向技術を導入しＲＤＢが進化，日経コンピュータ，日本，日経ＢＰ社，１９９６年１２月１９日，第４０６号，p.162-178

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F 12/00,17/30,13/00