(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第4378288号
(P4378288)

(45)発行日　平成21年12月2日(2009.12.2)　(24)登録日　平成21年9月18日(2009.9.18)

(51)Int.Cl.　　F I
G06F 21/24 (2006.01)　　G06F 12/14 560A
G06F 12/00 (2006.01)　　G06F 12/00 537Z

請求項の数 1　(全 24 頁)

(21)出願番号　特願2004-550039 (P2004-550039)
(86)(22)出願日　平成15年10月17日 (2003.10.17)
(65)公表番号　特表2006-505062 (P2006-505062A)
(43)公表日　平成18年2月9日 (2006.2.9)
(86)国際出願番号　PCT/US2003/033135
(87)国際公開番号　WO2004/043000
(87)国際公開日　平成16年5月21日 (2004.5.21)
審査請求日　平成18年10月16日 (2006.10.16)
(31)優先権主張番号　10/284,944
(32)優先日　平成14年10月31日 (2002.10.31)
(33)優先権主張国　米国(US)

早期審査対象出願

前置審査

(73)特許権者　390009531
インターナショナル・ビジネス・マシーンズ・コーポレーション
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＳＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国１０５０４　ニューヨーク州　アーモンク　ニュー　オーチャード　ロード
(74)代理人　100108501
弁理士　上野　剛史
(74)代理人　100112690
弁理士　太佐　種一
(74)代理人　100091568
弁理士　市位　嘉宏

最終頁に続く

(54)【発明の名称】データに対してセキュリティを実現する方法

(57)【特許請求の範囲】
【請求項１】
コンピュータにより、データに対するセキュリティを実現する方法であって、
コンピュータが、ユーザがデータベースに対して発行した照会を受領するステップと、
セキュリティに違反する傾向が存在するか否かを、前記コンピュータが、
（ｉ）前記ユーザが以前に発行した少なくとも１つの他の照会を基準にして前記照会を実行前に比較分析するステップ、および、
（ｉｉ）前記照会の実行から戻された結果と、前記以前に発行された少なくとも１つの他の照会の実行から戻された結果とを実行後に比較分析するステップ
のうちの少なくとも１つのステップに基づいて判断するステップとを有し、
セキュリティに違反する傾向が存在するか否かをステップ（ｉ）に基づいて判断する前記ステップが、さらに、前記照会と前記以前に発行された少なくとも１つの他の照会との間の相対共通度が所定値未満であるか否かを判断するステップと、
ＹＥＳの場合にセキュリティ規則を呼び出すステップとを備えた
方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は一般にデータ処理に関し、特に不適切なアクセスまたは無許可のアクセスからデータベースを保護する方法に関する。

【背景技術】
【０００２】
データベースとはコンピュータ化された、情報を記憶するとともに検索・取得するシステムのことである。リレーショナル・データベース管理システムとはデータを格納するとともに検索・取得するのにリレーショナル（関係型の）手法を使用する、コンピュータによるデータベース管理システム（database management sytem:ＤＢＭＳ）のことである。最も一般的な種類のデータベースはリレーショナル・データベース、すなわち多数の異なる方法で再構成しうるとともにアクセスしうるようにデータを定義する表形式のデータベースである。
【０００３】
特定のアーキテクチャとは無関係に、ＤＢＭＳにおいては、要求元のエンティティ（実体）（たとえばアプリケーションやオペレーティング・システムなど）がデータベース・アクセス要求を発行することにより特定のデータベースに対してアクセス権を要求する。そのような要求としては、たとえば単純なカタログ・ルックアップ（catalog lookup: カタログ探索）要求、または、データベース中の特定のレコードの読み取り、変更、もしくは付加を行うオペレーションを実行するトランザクション、および、それらトランザクションの組み合わせが挙げられる。これらの要求は高レベルの照会言語（たとえばＳＱＬ（Structured Query Language)）を用いて行う。例を用いて説明すると、ＳＱＬはたとえば次に示すようなデータベースから情報を取得したり、それを更新したりする対話型照会を作成するために使用する。すなわち、インターナショナル・ビジネス・マシーンズ・コーポレーション（International Business Machines Corporation)（ＩＢＭ(R)）製のＤＢ２(R)、マイクロソフト（Microsoft(R)）製のＳＱＬ　Ｓｅｒｖｅｒ、ならびに、オラクル（Oracle(R))、サイベース（Sybase(R))、およびコンピュータ・アソシエーツ（Computer Associates)製のデータベース製品などである。用語「照会（query)」は格納済みのデータベースからデータを検索・取得するためのコマンドから成る組を指している。照会はプログラマおよびプログラムにデータの選択、挿入、更新、所在の探索などを行わせるコマンド言語の形態をとる。
【０００４】
データベースの分野における重要な課題の１つがセキュリティである。データベースには多くの場合、機密データ、あるいは、それほどではないとしてもアクセスから保護するためにある程度のセキュリティを必要とする要注意データが含まれている。たとえば、医療記録は高度に個人的かつ機密であると考えられる。したがって、医療記録へのアクセスは通常、所定のユーザに限定されている。この目的のために、既存のデータベース管理システムでは権限のレベルを指定するユーザ・プロファイルを実装している。あるユーザがある特定のデータにアクセスしうるか否かは、その個別のプロファイル中に指定されている、当該ユーザの権限のレベルによって決まることになる。
【０００５】
しかし、上述した方法はきわめて硬直的であるとともに静的である。実際、そのような方法では、ユーザは妥当な範囲を超えたよりも広い範囲のデータにアクセスすることができない。その結果、データベースの有効性が著しく制限される。これに対して、セキュリティを緩和しすぎると、要注意データが危殆（きたい）に瀕（ひん）する。必要なことはデータのアクセス可能性とセキュリティとを均衡させることである。
【０００６】
既存のデータベースの弱点を説明するために、たとえば次に示す医療データベースを考える。すなわち、当該データベースに記録されている患者の匿名性を保証するために、ユーザが閲覧することを許可されている結果がクリニック（診療所）番号（clinic number)だけである医療データベースを考える。それでも、ユーザは特定の患者の既知の情報を用いて慎重に作成した一連の照会を発行することにより、当該患者のアイデンティティ（ＩＤ）を相当程度の確度で特定することができる。ここでは、そのようなプロセスを照会結合分析（query union analysis）と呼ぶ。次に示すのはクリニック番号（これはある個人

を一意に特定する１つの識別子である）と各照会が返す一意の患者の記録の個数とによって特定の個人を特定しうるように設計した一連の照会の例を示すものである。

【０００７】

照会	結果
１９９８年にアルツハイマーと診断された人々	１２００
カリフォルニア州で結婚して生活している人々	６０００
年齢が７０～８０歳で存命中の人々	１４０００
１９９９年と２００１年にクリニックを訪れたが、他の年には訪れなかった人々	６０００

【０００８】

独立に取り上げると、上記照会の各々は妥当な個数の結果を返している。しかし、全体としてみると、上記条件の各々を満たす結果の個数はきわめて少なくなり、おそらくたった１人になるであろう。ある個人用の１つのクリニック番号を特定したら、ユーザは複数のクリニック番号および他の任意の情報を返す何らかの照会を実行して、当該個人に対応する情報を特定することができる。

【０００９】

上述したのはユーザが既存のデータベースを利用する方法の一例にすぎない。他の様々な破壊的な（危険な）手法を用いて、データベースに格納されているデータを保護するために設けられているセキュリティ機構を迂回することができる。

【発明の開示】

【発明が解決しようとする課題】

【００１０】

したがって、データベース用の改良されたセキュリティ機構が求められている。

【課題を解決するための手段】

【００１１】

本発明は一般にデータベースのセキュリティのための方法、システム、および製品に関する。

【００１２】

一実例において、データに対するセキュリティを実現する方法を提供する。一実例は次に示すように構成する。

データに対するセキュリティを実現する方法であって、

ユーザがデータベースに対して発行した照会を受領するステップと、

セキュリティに違反する傾向が存在するか否かを、

（ｉ）前記ユーザが以前に発行した少なくとも１つの他の照会を基準にして前記照会を実行前に比較分析するステップ、および、

（ｉｉ）前記照会の実行から戻された結果と、前記以前に発行された少なくとも１つの他の照会の実行から戻された結果とを実行後に比較分析するステップ

のうちの少なくとも１つのステップに基づいて判断するステップと

を備えた

方法。

【００１３】

データに対するセキュリティを実現する別の方法は次に示すように構成する。

データに対するセキュリティを実現する方法であって、

ユーザから複数の照会を受領するステップと、

前記複数の照会をデータベースに対して実行するステップと、

前記ユーザが前記データベースに対して引き続いて発行した照会を受領するステップと、

前記複数の照会および前記引き続いて発行された照会に基づいて、前記データベース中

に存在するデータのうちの許可されていない量のデータにアクセススするユーザの試みが特定可能であるか否かをプログラムに基づいて判断するステップと
を備えた
方法。
【００１４】
データに対するセキュリティを実現する別の方法は次に示すように構成する。
データに対するセキュリティを実現する方法であって、
ユーザから複数の照会を受領するステップと、
前記複数の照会をデータベースに対して実行するステップと、
前記ユーザが前記データベースに対して引き続いて発行した照会を受領するステップと、
前記引き続いて発行された照会を実行するステップと、
前記複数の照会および前記引き続いて発行された照会に基づいて、セキュリティ制約を迂回して一意の本人確認を回避するユーザの試みが特定可能であるか否かをプログラムに基づいて判断するステップと
を備えた
方法。
【００１５】
特定の物理データ表記を有するデータに対するセキュリティを実現する別の方法は次に示すように構成する。
特定の物理データ表記を有するデータに対するセキュリティを実現する方法であって、
抽象照会を定義する複数の論理フィールドを備えた照会仕様を準備するステップと、
前記複数の論理フィールドを前記データの物理エンティティにマップするマッピング規則を準備するステップと、
セキュリティ規則を準備するステップと、
ユーザが前記データに対して発行した抽象照会を受領するステップであって、前記抽象照会は前記照会仕様に基づいて定義されているとともに、少なくとも１つの論理フィールドの値を用いて構成されている、ステップと、
前記ユーザから以前に受領した少なくとも１つの抽象照会を基準にして前記抽象照会を分析し、セキュリティ規則の呼び出しを引き起こす、セキュリティに違反するアクティビティの存在を探知するステップと
を備えた
方法。
【００１６】
さらに別の実例は次に示すように構成された、実行されると、セキュリティ違反を探知するオペレーションを行う命令群ををを格納したコンピュータ読み取り可能な媒体を提供する。
実行されると、セキュリティ違反を探知するオペレーションを行う命令群であって、
ユーザがデータベースに対して発行した照会を受領するステップと、
セキュリティに違反する傾向が存在するか否かを、
（ｉ）前記ユーザが以前に発行した少なくとも１つの他の照会を基準にして前記照会を実行前に比較分析するステップ、および、
（ｉｉ）前記照会の実行から戻された結果と、前記以前に発行された少なくとも１つの他の照会の実行から戻された結果とを実行後に比較分析するステップ
のうちの少なくとも１つのステップに基づいて判断するステップと
を備えた命令群
を格納した
コンピュータ読み取り可能な媒体。
【００１７】
さらに別の実例は次に示すように構成した、セキュリティ確認命令群を格納したコンピ

ュータ読み取り可能な媒体を提供する。
実行されると、
ユーザから複数の照会を受領するステップと、
前記複数の照会をデータベースに対して実行するステップと、
前記ユーザが前記データベースに対して引き続いて発行した照会を受領するステップと、
前記複数の照会および前記引き続いて発行された照会に基づいて、前記データベース中に存在するデータのうちの許可されていない量のデータにアクセススするユーザの試みが特定可能であるか否かをプログラムに基づいて判断するステップと
を備えた、
セキュリティを確認するオペレーションを行うセキュリティ確認命令群
を格納した
コンピュータ読み取り可能な媒体。
【００１８】
さらに別の実例は次に示すように構成した、その上に情報が格納されたコンピュータ読み取り可能な媒体を提供する。
その上に情報が格納されたコンピュータ読み取り可能な媒体であって、
前記情報が、
抽象照会を定義する複数の論理フィールドを備えた照会仕様と、
前記複数の論理フィールドを前記データの物理エンティティにマップするマッピング規則と、
複数のセキュリティ規則と、
ユーザが前記データに対して発行した抽象照会を受領することに対応して、セキュリティに違反するアクティビティを探知するオペレーションを行いうるように実行可能な実行時コンポーネントであって、
前記抽象照会は前記照会仕様に基づいて定義されているとともに、少なくとも１つの論理フィールドの値を用いて構成されており、
セキュリティに違反するアクティビティを探知する前記オペレーションが、
ユーザが前記データに対して発行した抽象照会を受領するステップであって、前記抽象照会は前記照会仕様に基づいて定義されているとともに、少なくとも１つの論理フィールドの値を用いて構成されている、ステップと、
前記ユーザから以前に受領した少なくとも１つの抽象照会を基準にして前記抽象照会を分析し、セキュリティ規則の呼び出しを引き起こす、セキュリティに違反するアクティビティの存在を探知するステップと
を備えている、
実行時コンポーネントと
を備えている、
コンピュータ読み取り可能な媒体。
【発明を実施するための最良の形態】
【００１９】
〔概要〕
本発明は一般にユーザが許可無くデータにアクセスしようとするのを探知するシステム、方法、および製品に関する。一般に、ある照会の実行によって返される結果について実行および／または分析を行う前に、当該照会について分析を行う。（「Ａおよび／またはＢ」は「ＡおよびＢ、Ａ、またはＢ」を表わす。）一実施形態では、セキュリティ違反が発生したかも知れないということを検出すると、セキュリティ上の処置を少なくとも１つ講じる。たとえば、一実施形態では、ユーザの照会を実行しない。別の実施形態では、当該イベントを記録し、および／または、当該イベントについて管理者に通知する。
【００２０】
一実施形態では、セキュリティ機能はデータの論理モデルの一部として実装する。この

論理モデルはその下にあるデータ・リポジトリの論理ビューを提供するデータ・リポジトリ抽象化層として実装する。このように、データを、当該データを物理的に表す特定の態様とは無関係にする。また、照会抽象化層も設けているが、それはデータ・リポジトリ抽象化層に基づくものである。抽象照会の、特定の物理的なデータ表記に対して使用しうる形態への変換は、実行時コンポーネントが行う。しかし、ここで説明した抽象化モデルは本発明の少なくとも１つの実施形態を実現するものであるが、当業者が理解しうるように、ここで提供する概念は抽象化モデルを用いなくとも実現することができ、それでも同一または類似の結果を得ることができる。

【００２１】

本発明の一実施形態はコンピュータ・システム（たとえば図１に示すとともに後述するコンピュータ・システム）で使用するプログラム製品として実現する。このプログラム製品のプログラムは実施形態の機能（たとえば、ここで説明する方法）を定義するものであるか、様々な信号担持媒体に格納することができる。信号担持媒体の例は次に示す情報を含んでいるが、それらに限定されない。すなわち、（ｉ）非書き込み可能型記憶媒体（たとえばコンピュータに内蔵された読み取り専用記憶装置〔たとえばＣＤ－ＲＯＭ駆動装置によって読み取り可能なＣＤ－ＲＯＭディスクなど〕など）に永続的に格納された情報、（ii）書き込み可能記憶媒体（たとえばディスケット駆動装置内のフロッピー(R) ディスクやハード・ディスク駆動装置など）に格納された変更可能な情報、または（iii)無線通信を含む通信媒体（たとえばコンピュータ・ネットワークまたは電話ネットワークなど）によってコンピュータに伝達される情報である。最後の実施形態には特に、インターネットおよび他のネットワークからダウンロードする情報が含まれる。このような信号担持媒体は、本発明の機能を指示するコンピュータ読み取り可能な命令群を担持しているときには、本発明の実施形態を表している。

【００２２】

一般に、本発明の実施形態を実現するために実行するルーチンはオペレーティング・システムもしくは特定のアプリケーションの一部、コンポーネント、プログラム、モジュール、オブジェクト、または命令群から成る手順である。本発明のソフトウェアは通常、ネイティブ・コンピュータ（native computer)が機械読み取り可能なフォーマットの、したがって実行可能な命令群に変換することになる複数の命令群から構成されている。また、プログラムは当該プログラムに限定的に存在する、またはメモリもしくは記憶装置に存在する変数およびデータ構造から構成されている。さらに、後述する様々なプログラムは本発明の特定の実施形態において当該プログラムを実装する際に対象とする用途に基づいて特定する。しかし、認識すべき点を挙げると、以下に示す特定の専門用語は単に便宜的に使用するだけである。したがって、本発明をそのような専門用語によって特定される、および／または示唆される特定の用途のみにおける使用に限定すべきではない。

【００２３】

〔環境の物理的な概観〕

図１は本発明の実施形態を実現しうるネットワーク接続されたシステム１００のブロック図である。一般に、ネットワーク接続されたシステム１００はクライアント（たとえばユーザの）コンピュータ１０２（このようなクライアント・コンピュータ１０２が３台示されている）および少なくとも１台のサーバ１０４（このようなサーバ１０４が１台示されている）を備えている。クライアント・コンピュータ１０２とサーバ・コンピュータ１０４とはネットワーク１２６を介して接続されている。一般に、ネットワーク１２６としてはＬＡＮ（local area network）および／またはＷＡＮ（wide area network)を用いることができる。特定の実施形態では、ネットワーク１２６はインターネットである。

【００２４】

クライアント・コンピュータ１０２は中央処理装置（Central Processing Unit:ＣＰＵ）１１０を備えている。ＣＰＵ１１０はバス１３０を介してメモリ１１２、記憶装置１１４、入力装置１１６、出力装置１１９、およびネットワーク・インタフェース装置１１８に接続されている。入力装置１１６としてはクライアント・コンピュータ１０２に入力を

供給する任意の装置を用いることができる。たとえばキーボード、キーパッド、ライトペン、タッチ・スクリーン、トラック・ボール、または、音声認識装置、音声／画像再生装置などを使用することができる。出力装置１１９としてはユーザに出力を供給する任意の装置（たとえば既存のディスプレイ画面など）を用いることができる。入力装置１１６から分離して示されているが、出力装置１１９と入力装置１１６は組み合わせることができる。たとえばディスプレイ画面とタッチ・スクリーンとを統合したもの、ディスプレイとキーボードとを統合したもの、音声認識装置とテキスト読み上げ変換装置とを統合したものなどを使用することができる。

【００２５】

ネットワーク・インタフェース装置１１８としてはクライアント・コンピュータ１０２とサーバ・コンピュータ１０４との間の、ネットワーク１２６を介したネットワーク通信を実現する任意の入口／出口（entry/exit）装置を用いることができる。たとえば、ネットワーク・インタフェース装置１１８としてはネットワーク・アダプタまたは他のネットワーク・インタフェース・カード（network interface card: ＮＩＣ）を用いることができる。

【００２６】

記憶装置１１４としては直接アクセス記憶装置（Direct Access Storage Device: ＤＡＳＤ）を用いるのが望ましい。それは単一の装置として示されているが、固定記憶装置および／または着脱可能な記憶装置を組み合わせたもの（たとえば固定ディスク駆動装置、フロッピー(R) ディスク駆動装置、テープ駆動装置、着脱可能メモリ・カード、光記憶装置など）を使用することができる。メモリ１１２と記憶装置１１４は複数の主記憶装置と二次記憶装置とにわたる１つの仮想アドレス空間の一部を構成していてもよい。

【００２７】

メモリ１１２としては本発明のプログラムとデータ構造を保持するのに十分な大きさを有するランダム・アクセス・メモリを用いるのが望ましい。メモリ１１２は単一の構成要素として示されているが、理解すべき点を挙げると、メモリ１１２は実際には複数のモジュールから成る。そして、メモリ１１２は高速のレジスタおよびキャッシュからより低速だがより大きなＤＲＡＭチップに至る複数の階層に存在する。

【００２８】

たとえば、メモリ１１２はオペレーティング・システム１２４を格納している。使用しうるオペレーティング・システムの例としてはＬｉｎｕｘ(TM)やマイクロソフト（Microsoft(R)）のＷｉｎｄｏｗｓ(R) が挙げられる。より一般的には、ここに開示する機能をサポートする任意のオペレーティング・システムを使用することができる。

【００２９】

また、メモリ１１２はブラウザ・プログラム１２２を格納しているようにも示されている。ブラウザ・プログラム１２２はＣＰＵ１１０で実行されると、様々なサーバ１０４の間をナビゲートすること、および、サーバ１０４のうちの少なくとも１つのもののネットワーク・アドレスを特定することを支援する。一実施形態では、ブラウザ・プログラム１２２はウェブを基にしたグラフィカル・ユーザ・インタフェース（ＧＵＩ）を備えている。これにより、ユーザはＨＴＭＬ（Hyper Text Markup Language）で記述された情報を表示することが可能になる。しかし、より一般的には、ブラウザ・プログラム１２２としてはサーバ・コンピュータ１０４から伝送されて来る情報を表示することのできる任意のプログラム（ＧＵＩを使用しているのが望ましい）を用いることができる。

【００３０】

サーバ・コンピュータ１０４はクライアント・コンピュータ１０２と同様の態様で物理的に構成されている。したがって一般に、サーバ・コンピュータ１０４はバス１３６によって相互に接続されたＣＰＵ１３０、メモリ１３２、および記憶装置１３４を備えているように示されている。メモリ１３２としてはサーバ・コンピュータ１０４に配置されたプログラムとデータ構造を保持するのに十分な大きさを有するランダム・アクセス・メモリを用いることができる。

【００３１】
一般に、サーバ・コンピュータ１０４はメモリ１３２に常駐しているように示されているオペレーティング・システム１３８の制御下にある。オペレーティング・システム１３８の例としてはＩＢＭ(R) ＯＳ／４００(R) 、ＵＮＩＸ(R) 、Ｍｉｃｒｏｓｏｆｔ(R) Ｗｉｎｄｏｗｓ(R) などが挙げられる。より一般的には、ここで説明する機能をサポートしうる任意のオペレーティング・システムを使用することができる。
【００３２】
メモリ１３２はさらに少なくとも１つのアプリケーション１４０と抽象照会インタフェース１４６とを備えている。アプリケーション１４０と抽象照会インタフェース１４６はコンピュータ・システム１００中の様々なメモリと記憶装置に様々な時に存在する複数の命令群を備えたソフトウェア製品である。アプリケーション１４０と抽象照会インタフェース１４６はサーバ１０４中の少なくとも１つプロセッサ１３０が読み込んで実行（execute)すると、本発明の様々な側面を具体化したステップ群すなわち構成要素群を実行（execute)するのに必要なステップ群をコンピュータ・システム１００に実行（perform)させる。アプリケーション１４０（ならびに、より一般的に任意の要求元エンティティ〔たとえばオペレーティング・システム１３８〕および〔最高のレベルに存在する〕ユーザ）はデータベース（たとえばデータベース156_1、・・・、156_N〔集合的にデータベース１５６として参照する〕）に対して照会を発行する。たとえば、データベース１５６は記憶装置１３４中に存在するデータベース管理システム（ＤＢＭＳ）の一部として示されている。データベース１５６はデータから成る任意の集合体を代表するものであり、特定の物理的な表現とは無関係である。たとえば、データベース１５６は（ＳＱＬによる照会によってアクセス可能な）リレーショナル・スキーマに従って、または、（ＸＭＬによる照会によってアクセス可能な）ＸＭＬスキーマに従って構成することができる。しかし、本発明は特定のスキーマに限定されず、現在は未知のスキーマへ拡張することが考えられる。ここで使用しているように、用語「スキーマ」は一般にデータの特定の配列を指す。
【００３３】
一実施形態では、アプリケーション１４０が発行する照会は各アプリケーション１４０に含まれているアプリケーション照会仕様１４２に従って定義する。アプリケーション１４０が発行する照会は事前に定義（すなわちアプリケーション１４０の一部としてハード・コード（hard code)化）しておいてもよいし、あるいは、入力（たとえばユーザ入力）に応答して生成してもよい。いずれの場合においても、照会（ここでは「抽象照会」と呼ぶ）は抽象照会インタフェース１４６によって定義されている論理フィールドを用いて作成／実行する。特に、抽象照会で使用する論理フィールドは抽象照会インタフェース１４６のデータ・リポジトリ抽象化コンポーネント１４８によって定義されている。抽象照会は実行時コンポーネント１５０が実行する。実行時コンポーネント１５０はまず、抽象照会を、ＤＢＭＳ１５４に含まれているデータの物理的な表記と一致する形態に変換する。
【００３４】
一実施形態では、データ・リポジトリ抽象化コンポーネント１４８はセキュリティ情報１６２を用いて構成する。抽象化モデルに基づかない実施形態（あるいは、その何らかの等価物）の場合、セキュリティ情報は他の場所に存在する。一実施形態では、セキュリティ情報１６２は少なくとも１つのフィールドに関連付けられたキーを含んでいる。そのようなキーの諸側面については下で詳細に説明する。
【００３５】
実行時コンポーネント１５０は様々な分析行うオペレーションを実行するが、一部の実施形態では実行した分析の結果に応じて様々なセキュリティ機能を執行したり、他の処置を講じたりする。したがって、実行時コンポーネント１５０は（複数のアルゴリズムを代表する）セキュリティ・アルゴリズム１５１（これは、ここで説明する方法を実装している）で構成されているように示されている。一般に、実行時コンポーネント１５０が実装しているセキュリティ機能は特定のユーザ、一群のユーザ、またはすべてのユーザに適用することができる。

【００３６】
一実施形態では、照会の構成要素はユーザがグラフィカル・ユーザ・インタフェース（ＧＵＩ）を用いて指定する。このＧＵＩのコンテンツはアプリケーション１４０が生成する。特定の実施形態では、このＧＵＩのコンテンツはＨＴＭＬ（hypertext markup language)で記述されたコンテンツであり、クライアントのコンピュータ・システム上に表示される。したがって、メモリ１３２にはクライアント・コンピュータ１０２が出す要求を処理するｈｔｔｐ（Hypertext Transfer Protocol)サーバ・プロセス１５２（たとえばウェブ・サーバ）が格納されている。たとえば、サーバ・プロセス１５２はデータベース１５６へのアクセスを求める要求に応答するものであり、たとえばサーバ１０４に常駐している。データベース１５６に存在するデータを求めて入来するクライアントの要求はアプリケーション１４０を呼び出す。アプリケーション１４０はプロセッサ１３０によって実行されると、サーバ・コンピュータ１０４に本発明の様々な側面（たとえばデータベース１５６にアクセスすることなど）を具体化したステップ群すなわち構成要素群を実行させる。一実施形態では、アプリケーション１４０は後刻、ブラウザ・プログラム１２２によって表示されることになるＧＵＩの構成要素を構築するように構成された複数のサーブレットを備えている。

【００３７】
図１はネットワーク接続されたクライアント・コンピュータ１０２とサーバ・コンピュータ１０４用のハードウェア／ソフトウェアの単なる一構成例にすぎない。本発明の実施形態群は、コンピュータ・システムが複雑かつマルチユーザ型のコンピューティング機器、単一ユーザ型のワークステーション、あるいは自分自身の不揮発性記憶装置を備えていないネットワーク機器であるか否かとは無関係に、同等の任意のハードウェア構成に適用することができる。また、理解すべき点を挙げると、ＨＴＭＬを含む特定のマークアップ言語を参照しているけれども、本発明は特定の言語、標準、またはバージョンに限定されない。したがって、当業者が理解しうるように、本発明は他のマークアップ言語および非マークアップ型の言語に適用可能であるし、また、本発明は特定のマークアップ言語の将来の変更および現在未知の他の言語にも適用可能である。同様に、図１に示すｈｔｔｐサーバ・プロセス１５２は単なる例示にすぎず、既知のプロトコルおよび未知のプロトコルをサポートする他の実施形態が考えられる。

【００３８】
〔環境の論理的な概観／実行時の概観〕
図２および図３は本発明のコンポーネント群の模式的な関係図２００を示す図である。要求元エンティティ（たとえばアプリケーション１４０の１つ）は当該要求元エンティティの個別アプリケーション照会仕様１４２によって定義した照会２０２を発行する。結果として得られる照会をここでは「抽象照会」と呼ぶ。というのは、ＤＢＭＳ１５４中に存在する、基となる物理的なエンティティを直接に参照するのではなく、抽象（すなわち論理）フィールドによって当該照会を作成しているからである。この結果、基となるデータに使用されている特定の表記とは無関係である抽象照会を定義することができる。一実施形態では、アプリケーション照会仕様１４２はデータを選択するために使用する基準（選択基準２１４）、および、選択基準２１４に基づいて戻すべきフィールドの明示的な仕様（戻りデータ仕様２０６）の双方を含んでいる。

【００３９】
図３に示す抽象照会２０２に対応する抽象照会の例を下のテーブルＩに示す。実例として、抽象照会２０２はＸＭＬを用いて定義されている。しかし、他の任意の言語を使用してもよい。

【００４０】
テーブルＩ－照会の例

```
001 <?xml version="1.0"?>
002 <!--Query string representation: (FirstName = "Mary" AND LastName =
003 "McGoon") OR State = "NC"-->
```

```
004 <QueryAbstraction>
005  <Selection>006   <Condition internalID="4" >
007    <Condition field="FirstName" operator="EQ" value="Mary"
008 internalID="1"/>
009    <Condition field="LastName" operator="EQ" value="McGoon"
010 internalID="3" relOperator="AND"></Condition>
011   </Condition>
012   <Condition field="State" operator="EQ" value="NC" internalID="2"
013 relOperator="OR"></Condition>
014  </Selection>
015  <Results>
016   <Field name="FirstName"/>
017   <Field name="LastName"/>
018   <Field name="State"/>
019  </Results>
020 </QueryAbstraction>
```

【００４１】

例示したように、テーブルＩに示す抽象照会は選択基準を含む選択（selection)仕様（第００５～０１４行）および結果（result）仕様（第０１５～０１９行）を含んでいる。一実施形態では、選択基準は（論理フィールドにおける）フィールド名、比較演算子（＝、＞、＜、など）、および（当該フィールドを比較する対象である）値式から成る。一実施形態では、結果仕様は照会を実行した結果として戻すことになる抽象フィールド群から成るリストである。抽象照会中の結果仕様はフィールド名と分類基準から成る。

【００４２】

アプリケーション照会仕様１４２が指示するとともに抽象照会２０２を作成するのに使用する論理フィールドは、データ・リポジトリ抽象化コンポーネント１４８が定義している。一般に、データ・リポジトリ抽象化コンポーネント１４８は、データを選択するための基準を指定するため、および、照会オペレーションが戻す結果データのフォーマットを指定するために、アプリケーション１４０が（ユーザが入力する照会の条件に応答して）発行する照会中で使用しうる情報を１組の論理フィールドとして公表している。これらの論理フィールド群はＤＢＭＳ１５４中で使用されている、基をなすデータ表記とは無関係に定義されているから、この基をなすデータ表記と緩やかに結合された照会を形成することが可能になる。

【００４３】

一般に、データ・リポジトリ抽象化コンポーネント１４８は複数のフィールド（field)仕様208_1、208_2、208_3、・・・（例として３つ示されている）（集合的にフィールド仕様２０８と呼ぶ）を備えている。特に、抽象照会を作成するのに利用可能な各論理フィールドは１つのフィールド仕様を備えている。一実施形態では、フィールド仕様２０８は論理フィールド名（name）210_1、210_2、210_3（集合的にフィールド名２１０）および関連するアクセス方式（access method)212_1、212_2、212_3（集合的にアクセス方式２１２）から成る。

【００４４】

アクセス方式２１２は論理フィールド名を、データベース（たとえばデータベース１５６のうちの１つ）中に存在する特定の物理データ表記214_1、214_2、・・・、214_Nに関連付けている（すなわちマップしている）。たとえば、図２には２つのデータ表記が示されている。すなわち、ＸＭＬデータ表記214_1とリレーショナル・データ表記214_2とである。しかし、物理データ表記214_Nは既知あるいは未知の他のすべてのデータ表記が考えられることを表している。

【００４５】

一実施形態では、単一のデータ・リポジトリ抽象化コンポーネント１４８が少なくとも

２つ物理データ表記用のフィールド仕様（および付随するアクセス方式）を含んでいる。別の実施形態では、個別の物理データ表記２１４ごとに異なる単一のデータ・リポジトリ抽象化コンポーネント１４８を備えている。さらに別の実施形態では、複数のデータ・リポジトリ抽象化コンポーネント１４８を備えている。この場合、各データ・リポジトリ抽象化コンポーネント１４８は（少なくとも１つの物理データ表記を有する、）基をなす同一の物理データの様々な部分を公表している。このように、複数のユーザが単一のアプリケーション１４０を同時に用いて、基をなす同一の物理データにアクセスすることができる。この場合、アプリケーションに公表される、基をなすデータの特定の部分は個別のデータ・リポジトリ抽象化コンポーネント１４８が決める。

【００４６】

サポートすべき様々な種類の論理フィールドの個数に応じて、多数のアクセス方式が考えられる。一実施形態では、単純フィールド用のアクセス方式、フィルタリング済みフィールド用のアクセス方式、および合成フィールド用のアクセス方式を備えている。フィールド仕様208_1、208_2、208_3はそれぞれ、単純フィールド用のアクセス方式212_1、212_2、212_3を示している。単純フィールドは基をなす物理データ表記中の特定のエンティティに直接にマップされている（たとえば所定のデータベースの表と列にマップされた１つのフィールド）。たとえば、図３に示す単純フィールド用のアクセス方式212_1は論理フィールド名（「FirstName 」）を「contact （接触）」なる名前の表(table) 中の「f __name」なる名前の列（column）にマップしている。フィルタリング済みフィールド（図２と図３には例示せず）は関連する物理的エンティティを特定するとともに、物理データ表記中の項目から成る特定のサブセットを定義するのに使用する規則を規定している。フィルタリング済みフィールドの一例としてニューヨーク州の郵便番号（ZIP code）フィールドが挙げられる。ニューヨーク州の郵便番号のフィールドは郵便番号の物理表記にマップされているとともに、データを、ニューヨーク州用に定義されたそれらの郵便番号だけに限定する。合成アクセス方式（図２と図３には例示せず）はアクセス方式の定義の一部として供給される式を用いて、少なくとも１つの物理フィールドから論理フィールドを算出する。このように、基をなすデータ表記中には存在しない情報を算出することができる。一例として、販売価格フィールドに売上税率を乗算して合成する売上税フィールドが挙げられる。

【００４７】

基をなすデータの所定の任意のデータ型（たとえば日付、１０進数、など）用のフォーマットは変化する可能性がある、と考えられる。したがって、一実施形態では、フィールド仕様２０８は基をなすデータのフォーマットを反映させた型属性を含んでいる。しかし、別の実施形態では、フィールド仕様２０８のデータ・フォーマットは関連する、基をなすデータとは異なっている。その場合、アクセス方式が、要求元エンティティが想定する適切なフォーマットでデータを戻す任に当たる。したがって、アクセス方式は基をなす物理データの実際のフォーマットに加え想定される（すなわち論理フィールドに従う）フォーマットを知っている必要がある。そうすれば、アクセス方式は基をなす物理データを論理フィールドのフォーマットに変換することができる。

【００４８】

たとえば、図２および図３に示すデータ・リポジトリ抽象化コンポーネント１４８のフィールド仕様２０８はリレーショナル・データ表記214_2で表したデータにマップされた論理フィールドを代表している。しかし、データ・リポジトリ抽象化コンポーネント１４８の他の例では、論理フィールドを他の物理データ表記（たとえばＸＭＬ）にマップしている。

【００４９】

一実施形態では、フィールド仕様２０８のうちの少なくとも１つは上で図１を参照して簡単に説明したセキュリティ情報１６２を用いて構成する。説明中の実施形態では、フィールド仕様208_3だけが関連するセキュリティ情報１６２を有する。したがって、すべてのフィールド定義が必ずセキュリティ情報を有する必要はない、ということを理解すべ

きである。現在の実施形態では、セキュリティ情報は値「key 」を有する型（type）属性である。理解すべき点を挙げると、このキーの値はデータ・リポジトリ抽象化１４８中で指定する必要はなく、その代わりに、たとえば構成ファイル中の値を用いてもよい。動作中、キーを有するとともにユーザが少なくとも１つの照会に含めたフィールドごとにセッション固有リスト１５３（図１にはこれが複数個示されている）を保持する。特に、リスト１５３（たとえばハッシュ・テーブル）は特定のセッションの間に関連するフィールドから戻された値をすべて含んでいる。したがって一般に、所定のユーザ用のサイズ・リストは以前に戻されていない結果（すなわち重複しない照会結果）を戻す各照会ごとに大きくなる。一実施形態ではこのリストは永続的であるが、別の実施形態ではユーザがログアウトする時、またはユーザの静止状態が一定期間継続した後にこのリストを削除する。次いで、照会結果の分析を行う。これについては、下で詳細に説明する。ある場合には、セキュリティ・アクション定義２１３に従ってアクションを実行する。アクションの例は下で説明する。

【００５０】

表ＩＩは図３に示すデータ・リポジトリ抽象化コンポーネント１４８に対応するデータ・リポジトリ抽象化コンポーネントの例を示すものである。たとえば、データ・リポジトリ抽象化１４８はＸＭＬを用いて定義する。しかし、他の任意の言語を用いてもよい。

【００５１】

テーブルＩＩ－データ・リポジトリ抽象化の例

```
<?xml version="1.0"?> <DataRepository>
 <Category name="Demographic">
  <Field queryable="Yes" name="FirstName" displayable="Yes">
     <AccessMethod>
       <Simple columnName="f__name" tableName="contact"></Simple>
     </AccessMethod>
     <Type baseType="char"></Type>
  </Field>
  <Field queryable="Yes" name="LastName" disdplayable="Yes">
     <AccessMethod>
       <Simple columnName="l__name" tableName="contact"></Simple>
     </AccessMethod>
     <Type baseType="char"></Type>
  </Field>
  <Field queryable="Yes" name="Clinic Number" displayable="Yes">
     <AccessMethod>
       <Simple columnName="CN" tableName="contact"></Simple>
     </AccessMethod>
     <Type baseType="char" key="true"></Type>
     <Security>
        <SecurityRule>
           <User>All</User>
           <Action>RunAndLog</Action>
        </SecurityRule>
        <SecurityRule>
           <User>securityOfficers</User>
           <Action>RunAndLog</Action>
        </SecurityRule>
        <SecurityRule>
           <User>cujo</User>
           <Action>NoAction</Action>
```

```
        </SecurityRule>
      </Security>
    </Field>
  </Category></DataRepository>
```

【００５２】

図４および図５は実行時コンポーネント１５０のオペレーションの一実施形態を例示する実行時の方法３００を示す図である。ステップ３０２において方法３００を開始する。その時、実行時コンポーネント１５０は抽象照会（たとえば図２に示す抽象照会２０２）のインスタンスを入力として受領する。ステップ３０４において、実行時コンポーネント１５０は当該抽象照会の当該インスタンスを読み取りかつ分析して個々の選択基準と所望の結果フィールドの所在を特定する。ステップ３０９において、何らかの事前のステートメント構造分析を行うが、それは実行後の結果分析のに際に効果的に使用されることになる。この点については後述する。特に、ステップ３０９において、照会共通度（commonality)値を算出する。照会共通度値は現在の照会と以前のすべての照会との間の相対共通度を求めることにより算出する。たとえば、ある照会が２つの条件（〔クニック番号＞ｘ〕かつ〔郵便番号＝ｙ〕）を有し、別の照会が同一のユーザについて２つの条件（〔クニック番号＜１０００〕かつ〔診断結果＝ｚ〕）を有する場合、これら２つの照会の共通度は５０％である。

【００５３】

ステップ３０６において、実行時コンポーネント１５０は抽象照会中に存在する各照会の選択基準ステートメントを処理するループに入ることにより、「具体照会（Concrete Query）」のデータ選択部分を構築する。一実施形態では、選択基準（ここでは条件とも呼ぶ）は（論理フィールド用の）フィールド名、比較演算子（＝、＞、＜など）、および当該フィールドの比較対象である値式から成る。ステップ３０８において、実行時コンポーネント１５０は抽象照会の選択基準中のフィールド名を用いてデータ・リポジトリ抽象化１４８中に存在する当該フィールドの定義を探索する。上述したように、フィールド定義は当該フィールドに付随する物理データにアクセスするのに使用するアクセス方式の定義を含んでいる。

【００５４】

ステートメント構造の分析を行うには、ステップ３１０から始めてさらに数ステップを要する。特に、ステップ３１０において、以前の照会の各々ごとにループに入る。すなわち、照会履歴テーブル１５７（図１）にアクセスして全探索を行う。一般に、照会履歴テーブル１５７は実行済みの照会から成るリストである。新たな照会を実行するたびに、照会履歴テーブル１５７に新たなエントリを追加する。一実施形態では、このデータ構造は抽象形態で表したＳＱＬ照会を含んでいる。このデータ構造は履歴を解放する時点について構成することができる。履歴を解放する選択肢の１つはセッションが終了する時である。別の選択肢は所定時間の経過後である。ステップ３１２において、実行時コンポーネント１５０は処理中（ステップ３０６）の照会選択のフィールドが、ステップ３１０において履歴照会テーブル１５７から検索・取得した以前の照会中で使用された否かを判断する。ＮＯの場合、方法３００はステップ３１０に戻り、実行時コンポーネント１５０が履歴照会テーブル１５７から別の以前の照会を検索・取得する。処理中（ステップ３０６）の照会選択のフィールドを有する以前の照会を特定したら、当該照会選択と特定した以前の照会とに対して分析を行う（ステップ３１４）。ステップ３１６において、実行時コンポーネント１５０は（ステップ３１４における）分析の結果として何らかのアクションを実行する必要があるか否かを判断する。一実施形態では、上記アクションはデータ・リポジトリ抽象化コンポーネント１４８中に指定されている（テーブルＩＩ参照）。セキュリティ・アクションにはユーザの照会（または他の適切な情報）を記録すること、照会が実行されるのを防止すること、および／または、ユーザのセッションを終了させることがある。より一般的には、当業者が認識しうるように、セキュリティ規則が呼び出されたら、様々な対応をすることができる。たとえば、（たとえば電子メールによる）システム管理者

への通知を発行する。留意点を挙げると、テーブルＩＩに示した例では、個々のユーザ（たとえばCujo）ごと、ユーザ達から成るグループ（たとえばセキュリティ担当者達）ごと、および、すべてのユーザごとにセキュリティ・アクションを定義している。一実施形態では、特定のフィールド用に複数のアクションが存在する場合、あるユーザにとって最狭義に調製されたアクションを適用する。したがって、個々のユーザに特有のアクションは他のすべてのアクションに優先し、あるグループに特有のアクションはすべてのユーザ用に指定されているアクションに優先する。すべてのユーザ用に指定されているアクションが適用されるのは、特定のユーザ用により狭義に調製されたアクションが他に存在しない場合だけである。ステップ３１６の結果がＮＯの場合（すなわちアクションを必要としない場合）、プロセスはステップ３１０に戻り、そこで履歴照会テーブル１５７から別の以前の照会を検索・取得して検査の用に供する。ステップ３１６においてアクションを必要とする場合には、ステップ３１８において当該アクションを実行する。アクションが致命的である場合（ステップ３２０）、当該ユーザの照会は実行しない（ステップ３２２）。そうでない場合、プロセスはステップ３１０に戻る。履歴照会テーブル１５７中の以前の照会の各々について、処理中の現在の照会選択のフィールドが存在するか否かの検査が完了したら、方法３００はステップ３２４に戻る。

【００５５】

次いで、実行時コンポーネント１５０は処理中の論理フィールド用の「具体照会寄与部（Concrete Query Contribution)」を構築する。ここで定義しているように、「具体照会寄与部」は現在の論理フィールドに基づいてデータの選択を実行するために使用する具体照会の一部分である。具体照会とはＳＱＬや「ＸＭＬ　Ｑｕｅｒｙ」のような言語で表した照会のことであり、所定の物理データ・リポジトリ（たとえばリレーショナル・データベースやＸＭＬリポジトリなど）のデータと一致している。したがって、具体照会は図１に示すＤＢＭＳ１５４によって代表される物理データ・リポジトリ中のデータの所在を特定し、当該データを検索・取得するために使用する。次いで、現在のフィールド用に生成した「具体照会寄与部」を「具体照会文（Concrete Query Statement）」に付加する。次いで、方法３００はステップ３０６に戻り、抽象照会の次のフィールド用の処理を開始する。したがって、ステップ３０６において入ったプロセスは抽象照会中の各データ選択フィールドごとに繰り返す。これにより、実行すべき最終的な照会に内容が付加される。

【００５６】

具体照会のデータ選択部分を構築したら、実行時コンポーネント１５０は照会を実行した結果として戻すべき情報を特定する。上述したように、一実施形態では、抽象照会は照会を実行した結果として戻すべき抽象フィールドから成るリスト（ここでは結果仕様と呼ぶ）を定義している。抽象照会に設けた結果仕様はフィールド名と分類基準とから成る。したがって、方法３００はステップ３２８において（ステップ３２８、３３０、３３２、３３４で定義された）ループに入り、生成中の具体照会に結果フィールド定義を付加する。ステップ３３０において、実行時コンポーネント１５０は結果フィールド名を求めてデータ・リポジトリ抽象化１４８（中の抽象照会の結果仕様）を探索し、データ・リポジトリ抽象化１４８から「結果フィールド定義」を検索・取得して現在の結果フィールド用に戻すべきデータの物理的な場所を特定する。次いで、実行時コンポーネント１５０は論理結果フィールド用に（、戻すべきデータの物理的な場所を特定している具体照会の）「具体照会寄与部」を構築する（ステップ３３２）。次いでステップ３３４において、「具体照会寄与部」を「具体照会文」に付加する。抽象照会中に存在する結果仕様の処理が完了したら、ステップ３３６において当該照会を実行する。

【００５７】

ステップ３１０とステップ３１８によって論理フィールド用の「具体照会寄与部」を構築する方法４００の一実施形態を図６を参照して説明する。ステップ４０２において、方法４００は現在の論理フィールドに付随するアクセス方式が単純アクセス方式であるか否かを調べる。ＹＥＳなら、物理データの所在情報に基づいて「具体照会寄与部」を構築する（ステップ４０４）。次いで、上述した方法３００に従って処理を継続する。ＮＯなら

、処理はステップ４０６に進み、現在の論理フィールドに付随するアクセス方式がフィルタリング済みアクセス方式であるか否かを調べる。ＹＥＳなら、何らかの物理データ・エンティティ用の物理データの所在情報に基づいて「具体照会寄与部」を構築する（ステップ４０８）。ステップ４１０において、上記物理データ・エンティティに付随するデータをサブセット化するのに使用する追加の論理（フィルタの選択）を用いて「具体照会寄与部」を拡張する。次いで、上述した方法３００に従って処理を継続する。

【００５８】

当該アクセス方式がフィルタリング済みアクセス方式でない場合、処理はステップ４０６からステップ４１２に進む。そこで、方法４００は当該アクセス方式が合成アクセス方式であるか否かを調べる。当該アクセス方式が合成アクセス方式である場合、合成フィールド式中の各サブフィールド参照ごとに物理データの所在を特定し、それを取得する（ステップ４１４）。ステップ４１６において、合成フィールド式の論理フィールド参照に合成フィールド式の物理フィールドの場所情報を代入する。これにより、「具体照会寄与部」を生成される。次いで、上述した方法３００に従って処理を継続する。

【００５９】

当該アクセス方式が合成アクセス方式ではない場合、処理はステップ４１２からステップ４１８に進む。ステップ４１８は本発明の実施形態として考えられる他のすべてのアクセス方式の種別を代表している。しかし、理解すべき点を挙げると、利用可能なアクセス方式をすべては実装しない実施形態が考えられる。たとえば、特定の実施形態では、単純アクセス方式しか使用しない。別の実施形態では、単純アクセス方式とフィルタリング済みアクセス方式だけを使用する。

【００６０】

上述したように、論理フィールドが、基をなす物理データと異なるデータ・フォーマットを指定している場合、データの変換を行う必要がある。一実施形態では、論理フィールド用の「具体照会寄与部」を構築するときに個別のアクセス方式ごとに方法４００に従い最初の変換を行う。たとえば、この変換はステップ４０４、４０８、４１６の一部として、あるいはこれらのステップの直後に行う。物理データのフォーマットから論理フィールドのフォーマットへの次の変換はステップ３２２において照会を実行した後に行う。無論、論理フィールド定義のフォーマットが基をなす物理データと同一である場合、変換は必要ない。

【００６１】

図７を参照する。図７はステップ３１４において行う分析の一実施形態を説明する方法５００を示す図である。この分析は<field><operastor><value> なる一般的な様式を有する選択／条件に基づいて行う、という点を想起されたい。ステップ５０２において、演算子と値を用いて、照会の選択が対象とする範囲を決める。ステップ５０４において、実行時コンポーネント１５０はステップ３１０において履歴照会テーブル１５７から検索・取得した以前の照会の条件について非重複条件があるか否かを検査する。一実施形態では、非重複条件を、先行する照会の共通フィールドを備えているが、先行する照会が戻す結果（行）をまったく戻さない条件として定義している。たとえば、範囲条件「age>=0 AND age<5（年齢が０歳以上、かつ、５歳未満）」を有する以前の照会（その条件は履歴照会テーブル１５７に格納されている）を考える。いま、分析中の照会が範囲条件「age>=5 AND age<10 （年齢が５歳以上、かつ、１０歳未満）」を含んでいる、と仮定する。これらの照会条件は、ユーザは同一の行をまったく戻さないように設計された照会を意識的に作成することによりデータベースの大部分を走査するつもりである、ということを示唆する傾向（pattern)を明示している。別の実施形態では、非重複条件を、先行する照会の共通フィールドを備えており、いくつかの新しい結果（すなわち以前の照会が戻さない結果）といくつかの古い結果（すなわち以前の照会が戻した結果）とを戻す条件として定義している。このような非重複条件を繰り返す傾向はデータベースの一部分にアクセスし、それらを収集する無許可の試みとして特定することもできる。

【００６２】

非重複条件を特定したら、ステップ３１６／ステップ３１８において当該条件を処理する。一実施形態では、非重複条件は管理者の設定に従って処理する。特に、何らかのアクションを実行する前に特定する必要のある、関係のない照会の個数は管理者の設定によって指定する。また、一実施形態では、条件がある程度、重複または分離するのを許容している。したがって、共通する結果の個数がわずかである、２つの照会の条件群は非重複であると考えられる。そのような場合、非重複の判断は異なる照会群の条件群が対象にしている範囲に基づいて行うのが望ましい。たとえば、関係する１つのフィールドを有する照会群から成るあるグループの結果群の合計の範囲が４０００であり、条件群によって戻される重複する結果群の実際の個数が４である場合、当該照会群／条件群は実質的に非重複である。他方、関係する１つのフィールドを有する照会群から成るあるグループの結果群の合計の範囲が４０であり、照会群によって戻される重複する結果群の個数が３０である場合、当該照会群／条件群は実質的に重複していると考えられる。特許請求の範囲の解釈にあたっては、用語「非重複」の照会／条件は実質的に非重複の照会／条件を含むように解釈すべきである。さらに（または、あるいは）、結果を戻す対象をなす様々な患者の人数は管理者の設定によって定義する。一実施形態では、そのような管理者の設定は特定のユーザに特有なものにしている。したがって、第１のユーザにはデータに対してより大きなアクセス権が付与されるが、第２のユーザのアクセス権は比較的限定されたものになる、ということがありうる。

【００６３】

以上は実行前分析を例示している。追加または別の側面として、図５のステップ３３６における、照会の実行に続く実行後分析がある。実行後分析の例はブック３３８、３４０、３４２によって示されている。一般に、実行後分析には照会の実行後であって、実行した照会の結果をユーザに戻す前または後に行う処理が含まれる。たとえば、ブロック３３８は当該結果をユーザに提示する前に行う非重複の照会の分析を示している。ブロック３３８の非重複の照会の分析を行う方法６００の一実施形態を図８に示す。まず、実行時コンポーネント１５０はステップ６０２においてループに入る。ここのループは当該結果の各列ごとに実行する。ステップ６０４において、実行時コンポーネント１５０は当該列がキー列（すなわちキーが定義されている列）であるか否かを判断する。ＮＯの場合、当該結果の次の列を同様に処理する。当該結果がキー列を含んでいる場合、当該キー列に対応するリスト１５３の現在のサイズを検索・取得する（ステップ６０６）。リスト１５３に含まれていない結果の各値をリスト１５３に付加する（ステップ６０８）。ステップ６１０において、実行時コンポーネント１５０は非重複の照会が特定されているか否かを判断する。説明中の実施形態では、ステップ６１０には新たな値の各々を付加した（ステップ６０８）後の当該キー・リストのサイズが新たな結果／値と（ステップ６０６で検索・取得した）当該リストの元のサイズとの合計に等しいか否かの判断が含まれている。この点における肯定的な判断は、当該照会によって戻され、リスト１５３に付加した新たな値はない（その場合、ステップ３３６で実行した照会は以前の照会と重複していない）、ということを示している。

【００６４】

実行前分析について上述したように、ある程度の重複があっても、場合によっては実質的に非重複であると考えることができる。この原理は実行後分析に適用することができる。したがって、共通する結果の個数がわずかである、２つの照会の結果群は非重複であると考えられる。そのような場合、非重複の判断は戻される結果の合計個数に基づいて行うのが望ましい。たとえば、関係する１つのフィールドを有する照会群から成るあるグループの結果群の合計が４０００であり、重複する結果群の個数が４である場合、当該照会群は実質的に非重複である。他方、関係する１つのフィールドを有する照会群から成るあるグループの結果群の合計が４０であり、重複する結果群の個数が３０である場合、当該照会群は実質的に重複していると考えられる。特許請求の範囲の解釈にあたっては、用語「非重複」の照会／結果は「実質的に」非重複の照会／結果を含むように解釈すべきである。ステップ３３６において実行した照会が重複または実質的に重複であると判断されたら

、その結果にマークを付けてユーザへの戻しに備える（ステップ６１１）。そうでない場合、実行時コンポーネント１５０は事前に決めた何らかのアクション（この例は上述した）が必要であるか否かを判断する（ステップ６１４）。ＹＥＳなら、ステップ６１６において当該アクションを実行する。当該アクションが致命的である（とステップ６１８において判断した）場合、当該要求を終了し、ユーザには結果を戻さない（ステップ６２０）。次いで、方法６００は終了する。当該アクションが致命的ではない場合、プロセスはステップ６０２に戻り、そこで次の列の処理を開始する。致命的なアクションを呼び出すことなくすべての列の処理に成功したら、ユーザにすべての結果を戻す（ステップ６１２）。

【００６５】

非重複の照会を特定する実行後の照会分析の一例として、１０００個の異なるクリニック番号を戻す第１の照会を実行するユーザを考える。１０００個の異なるクリニック番号はクリニック番号用の適切なキー・リスト１５３を用いて追跡する。次いで、当該ユーザは１５００個の異なるクリニック番号を戻す第２の照会を実行する。第１の照会と第２の照会が完全に一意の結果を戻すものと仮定すると、クリニック番号用のキー・リスト１５３は２５００個の異なるクリニック番号を保持することになるから、上記照会群は非重複であると判断される。上記照会群が戻す結果群が共通の値を少なくとも１つ共有している場合、（上述したように）上記照会群が実質的に非重複であるか否かを判断するステップ群を実行する。より一般的には、様々な構成可能な設定を使用して非重複の照会の傾向を探知して致命的なアクションを過早に実行しないようにする（すなわち結果をユーザに戻さないようにする）。たとえば、アクションを実行する前に戻すことのでる非重複のキー値の個数を事前に定義しておく。あるいは（または、さらに）、アクションを実行する前に実行することのでる非重複の、または実質的に非重複の照会の個数を事前に定義しておく。当業者が認識しうるように、他の規則を使用してもよい。

【００６６】

留意点を挙げると、事前に定義しておくキーを使用することは様々な種類の照会の分析を行うための一実施形態にすぎない。より一般的には、照会間の共通性を追求することを可能にする任意の方法が考えられる。たとえば、事前にキーを定義しておくことの代案として、同一のユーザによる一連の照会を検査して共通のフィールドの存在を確認することが挙げられる。次いで、当該共通のフィールドを指定し、それを傾向の分析（たとえば非重複の照会の探知）を行う際に依拠するキーとして使用する。

【００６７】

別の種類の実行後照会分析は図５においてブロック３４０で表されている。ここでは、それを照会結合（union)分析による探知方式と呼ぶ。照会結合分析の一例を上で提示した。一般に、照会結合分析による探知方式では、一連の照会群を検査して、明らかに未接続である（すなわち異なる複数の条件から成る）にもかかわらず、結果の組を小さくしても少なくとも１つの共通の結果値を有する照会群の傾向を探知する。探知した後、照会結合分析を扱う一実施形態が図９に示す方法７００である。方法７００は照会の実行に続いて開始する。まず、ステップ７０２において、セキュリティ・アルゴリズム１５１が照会の結果を追跡する結果リストが存在するか否かを判断する。ＮＯなら、結果リスト１６１を作成し、結果をそこに格納する（ステップ７０４）。次いで、方法７００は終了する。しかし、結果リストが既に存在する場合、アルゴリズム１５１は結果リスト１６１から非共通の値をすべて破棄するオペレーションを実行する。すなわち、結果リスト１６１に格納されているが当該照会を実行することにより戻された結果の一部ではないすべての値を結果リスト１６１から除去する。ステップ７０８において、アルゴリズム１５１は結果リストのサイズがサイズ閾値（いきち、しきい値）（一実施形態では、サイズ閾値はカスタマイズ可能である）未満になっているか否かを判断する。ＮＯなら、ユーザに結果を戻した後（ステップ７１０）、方法７００は終了する。ＹＥＳなら、アルゴリズム１５１は（図４のステップ３０５において求めた）共通度値が共通度閾値未満であるか否かを判断する（ステップ７１２）。ＮＯなら、ユーザに結果を戻した後（ステップ７１０）、方法７０

０は終了する。ＹＥＳなら、事前に定義しておいたセキュリティ・アクションを実行する（ステップ７１４）。当該セキュリティ・アクションが致命的である（とステップ７１６において判断した）場合、当該ユーザの要求を停止させた後、方法７００は終了する。当該セキュリティ・アクションが致命的ではない場合、ユーザに結果を戻した後（ステップ７１０）、方法７００は終了する。

【００６８】

別の種類の実行後の照会分析は図５においてブロック３４２で表されている。ここでは、それを削減（pare down:切り詰め）分析による探知方式と呼ぶ。削減分析とは比較的大量の行を戻す広い照会を実行した後、最初の結果群を後続する照会群を用いて継続的かつ体系的にサブセット（部分集合）化するプロセスのことである。ある面では、削減分析は結合照会分析の変形例である。すなわち、両方法ともユーザにとって既知の情報を有効に利用して、戻される結果群のサイズを限定している。いま、アルツハイマーに罹（かか）っている人々を求める第１の照会を発行するユーザを考える。この第１の照会を実行することにより戻された結果を見て、当該ユーザは上記照会をカリフォルニア州に住んでいるような人々に限定することにより、より大きな特定性を達成できると判断する。したがって、アルツハイマーに罹り、かつカリフォルニア州に住んでいる人々を求める第２の照会を、上記ユーザは発行する。次いで、上記ユーザはさらに上記照会を特定の年齢の人々に限定する。上記ユーザは戻される結果の個数を少なくするために、このサブセット化する傾向を多数の照会にわたって継続する。

【００６９】

図１０は実行後に行う削減探知の方法８００の一実施形態を示す図である。方法８００は照会を実行し結果を受領した後に開始する。まず、ステップ８０４において、実行時コンポーネント１５０が、結果の個数が追跡閾値未満であるか否かを判断する。たとえば、追跡閾値は削減探知を行うべき時に基づいて選定した事前定義の値である。すなわち、ユーザに、ある程度の検索権限を与えるために、結果の個数が追跡閾値を超えている場合には削減探知を行わない。したがって、ステップ８０４の結果がＮＯの場合、照会を実行した結果をユーザに戻す（ステップ８０６）。しかし、結果の個数が追跡閾値未満である場合、実行時コンポーネント１５０は削減探知方法を前回呼び出してから少なくとも１つの結果リスト１６１（図１）が既に存在するか否かを判断する。一般に、削減探知を行いうるようにするために、結果リスト１６１には実行済みの照会の結果が含まれている。（ステップ８０８において）結果リストがまだ存在しない場合、現在の結果を結果リスト１６１に格納した後（ステップ８１０）、ユーザに戻す（ステップ８０６）。少なくとも１つの結果リスト１６１が存在する場合、実行時コンポーネント１５０は現在の結果リストが既存の結果リスト群のうちの任意の１つのもののサブセットであるか否かを判断する（ステップ８１２）。ＮＯなら、現在の結果を別の結果リストに格納する（ステップ８１４）。したがって、複数の結果リストが存在する可能性がある。その場合、各結果リストは複数の照会に対して戻された互いに関係のない結果の組を含んでいる。しかし、現在の結果が既存の結果リスト群のうちの１つのもののサブセットである場合には削減の傾向を探知済みであるから、セキュリティ・アクションを呼び出す（ステップ８１６）。セキュリティ・アクションの例については上述した。当該セキュリティ・アクションが致命的である（とステップ８１８において判断した）場合にはユーザに現在の結果を戻さないから、当該ユーザはさらなる照会を実行することができない（ステップ８２０）。当該セキュリティ・アクションが致命的ではない場合にはユーザに現在の結果を戻す（ステップ８０６）。

【００７０】

上述した削減法８００では、削減の傾向を探知しうるのは２つの照会を実行した後であって、しかも双方の結果の個数が追跡閾値未満である（とステップ８０４において確認された）ものと見なしうる場合だけである。しかし、理解すべき点を挙げると、削減の傾向を探知するための個々の基準は構成可能である。たとえば、上記削減アルゴリズムでは、（結果の個数が追跡閾値未満であることに加え）削減の傾向がＮ（Ｎ＞２）個の照会にわ

たる必要がある。また、上記削減アルゴリズムでは、削減の傾向が順次／連続した照会群にわたって生じる必要がある。当業者が認識しうるように、他の基準を用いてもよい。

【００７１】

一実施形態では、「ホット・リスト」を使用してもよい。このホット・リストにはより高度のセキュリティを受けるのに値する所定の個人が含まれている。一実施形態では、当該ユーザとは無関係に、すべてのユーザ用に単一のホット・リストを使用する。このような方法が有益なのは、ホット・リストに列挙されている個人が著名人の場合である。別の実施形態では、ホット・リストに各ユーザにとって既知の個人が含まれるように、当該ホット・リストを各ユーザに合わせてパーソナライズしている。このように、特定のユーザのホット・リストに搭載されている少なくとも１人の個人を指向する当該特定のユーザによる検索を、匿名性と機密性を保ちながら探知し、処理することができる。

【００７２】

上述したように、データ・リポジトリ抽象化コンポーネント１４８は様々な利点をもたらす一実施形態の単なる例示である。ある面では、アプリケーションによる照会の指定と基をなすデータの表記との間に緩やかな結合を定義すると好都合である。つまり、特定の表、列、および関係情報を用いてアプリケーションをエンコードするのではなく、ＳＱＬを使用する場合と同様に、アプリケーションには、データ照会の要件を、後ほど実行する時に特定の物理データ表記に結び付けるという、より抽象的な態様で定義する。本発明では照会－データ間の結合が緩やかであるから、要求元エンティティ（たとえばアプリケーション）は、たとえ基をなすデータの表記を変更しても、あるいは、要求元エンティティを開発した時に使用した物理データ表記を完全に新しくした物理データ表記によって当該要求元エンティティを使用すべき場合であっても機能することが可能になる。所定の物理データ表記を変更または再構成する場合には、対応するデータ・リポジトリ抽象化を更新して、基をなす物理データ・モデルになされた変更を反映させる。以前と同じ組の論理フィールドが照会による使用の際に利用可能である。それらは物理データ・モデル中の異なるエンティティまたは場所に結び付けられているにすぎない。この結果、抽象照会インタフェースに書き込まれた要求元エンティティは、たとえ対応する物理データ・モデルが大幅に変更されても、不変のまま機能し続ける。要求元エンティティを開発した時に使用した物理データ表記を完全に新しくした物理データ表記によって当該要求元エンティティを使用すべき場合には、同じ技術（たとえばリレーショナル・データベース）を使用するが、異なる命名と情報の構成の戦略（たとえば異なるスキーマ）に従って新たな物理データ・モデルを実装する。新たなスキーマは単純アクセス方式、フィルタリング済みアクセス方式、および合成フィールド・アクセス方式の手法を使用するアプリケーションが必要とする論理フィールドの組にマップする情報を含むことになる。あるいは、新たな物理表記では、類似の情報を表記する別の技術を使用する（たとえば、リレーショナル・データベース・システムに対してＸＭＬに基づいたデータ・リポジトリを使用する）。いずれにしても、照会中で参照されるフィールドを新たな物理データ・モデル中の場所と物理表記にマップする別のデータ・リポジトリを備えることにより、抽象照会インタフェースを使用するように書かれている既存の要求元エンティティは新たな物理データ表記の使用に容易に移行することができる。

【００７３】

エンド・ユーザについては、データ・リポジトリを抽象化することにより、データ・フィルタリング機構、適切なデータの公開、および所定のコンテンツに対するアクセスの防止が実現する。しかし、データ・リポジトリを抽象化することは本発明の単なる一実施形態にすぎない、という点を理解すべきである。より一般的には、本発明はユーザ－データ間の従属性に従って照会を実行する（あるいは実行しない）任意の態様で実現することができる。すなわち、照会の実行を行うか否かはエンド・ユーザと、実行時に照会がアクセスしたり戻したりする特定のデータとに従属している。

【００７４】

しかし、強調すべき点を挙げると、当業者が容易に認識しうるように、本発明のセキュ

リティ機能とセキュリティ機構はデータ・リポジトリ抽象化コンポーネントとは切り離して実装することができる。たとえば、伝統的なリレーショナル・データベースの状況おいて、一実施形態では構造体群を照会パーサから離して使用しているが、それらはデータベース・エンジンに常駐して、ここで説明した分析を実行することもできる。

【００７５】

上述した事項は本発明の実施形態に関するが、本発明の基本的な範囲の内で本発明の他の実施形態とさらなる実施形態とを案出することができる。そして、本発明の範囲は特許請求の範囲によって確定される。

【図面の簡単な説明】

【００７６】

【図１】コンピュータ・システムの一実施形態を示す図である。

【図２】本発明の一実施形態のソフトウェア・コンポーネントの論理的／物理的な概観を示す図である。

【図３】抽象照会の論理的な概観と抽象化のデータ・リポジトリとを示す図である。

【図４】実行時コンポーネントのオペレーションを示すフローチャートを示す図である。

【図５】実行時コンポーネントのオペレーションを示すフローチャートを示す図である。

【図６】実行時コンポーネントのオペレーションを示すフローチャートを示す図である。

【図７】実行前分析を用い非重複条件を探知して処理する実行時コンポーネントのオペレーションを示すフローチャートを示す図である。

【図８】実行後結果分析を用い非重複条件を探知して処理する実行時コンポーネントのオペレーションを示すフローチャートを示す図である。

【図９】実行後結果分析を用い照会結合分析を探知して処理する実行時コンポーネントのオペレーションを示すフローチャートを示す図である。

【図１０】実行後結果分析を用い削減分析を探知して処理する実行時コンポーネントのオペレーションを示すフローチャートを示す図である。

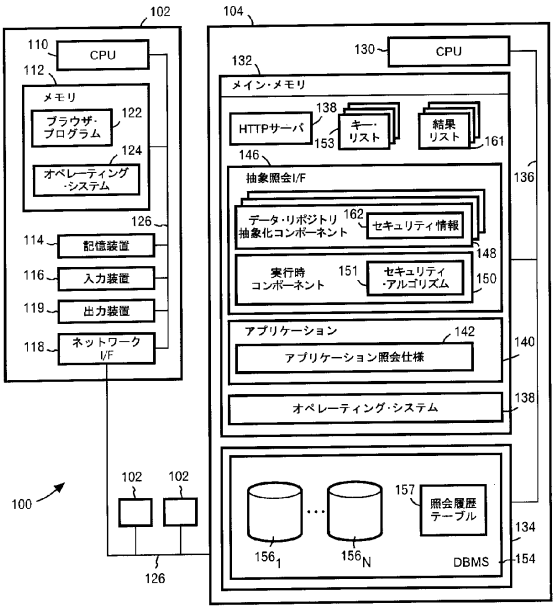
【符号の説明】

【００７７】

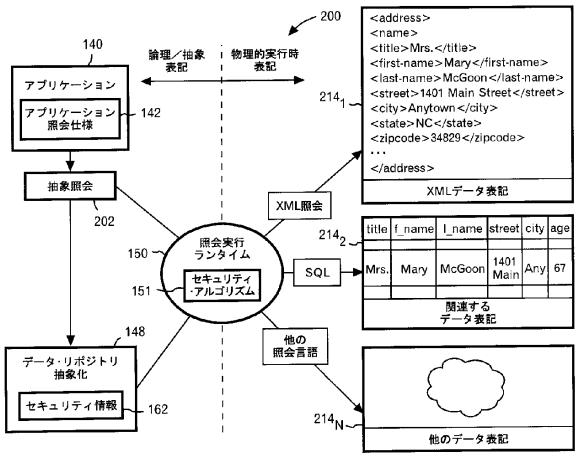
１００　システム
１０２　クライアント・コンピュータ
１０４　サーバ・コンピュータ
１１０　ＣＰＵ
１１２　メモリ
１１４　記憶装置
１１６　入力装置
１１８　ネットワークＩ／Ｆ
１１９　出力装置
１２２　ブラウザ・プログラム
１２４　オペレーティング・システム
１２６　ネットワーク
１３０　ＣＰＵ
１３２　メイン・メモリ
１３４　記憶装置
１３６　バス
１３８　ＨＴＴＰサーバ
１３８　オペレーティング・システム
１４０　アプリケーション
１４２　アプリケーション照会仕様
１４６　抽象照会Ｉ／Ｆ
１４８　データ・リポジトリ抽象化コンポーネント
１５０　実行時コンポーネント

１５１　セキュリティ・アルゴリズム
１５３　キー・リスト
１５４　ＤＢＭＳ
１５６　データベース
１５７　照会履歴テーブル
１６１　結果リスト
１６２　セキュリティ情報
２００　コンポーネント群の模式的な関係図
２０２　抽象照会
２１４　選択基準

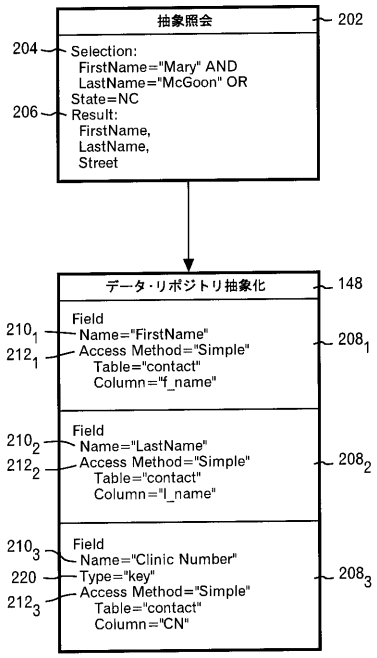
【図１】



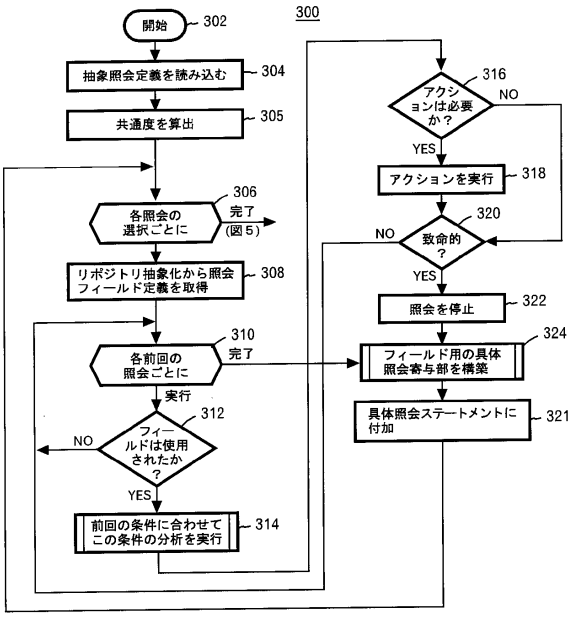
【図２】



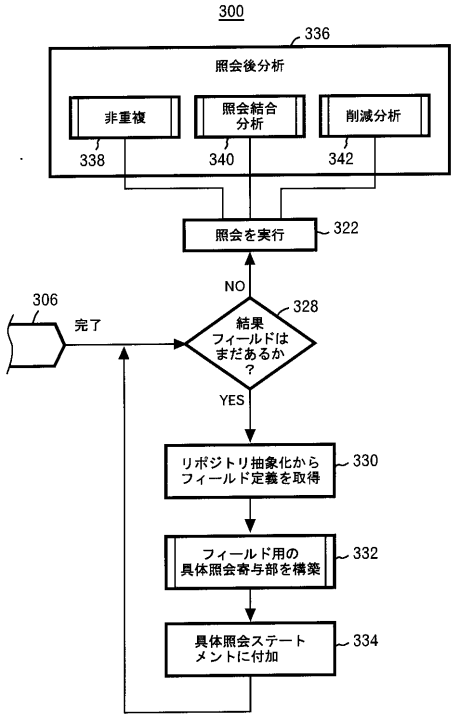
【図３】


抽象照会
202
204
Selection:
FirstName="Mary" AND
LastName="McGoon" OR
State=NC
206
Result:
FirstName,
LastName,
Street
データ・リポジトリ抽象化
148
Field
Name="FirstName"
Access Method="Simple"
Table="contact"
Column="f_name"
210_1
212_1
208_1
Field
Name="LastName"
Access Method="Simple"
Table="contact"
Column="l_name"
210_2
212_2
208_2
Field
Name="Clinic Number"
Type="key"
Access Method="Simple"
Table="contact"
Column="CN"
210_3
220
212_3
208_3

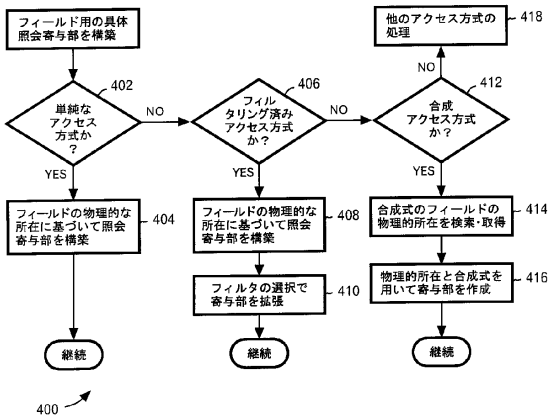
【図４】


300
開始
302
抽象照会定義を読み込む
304
共通度を算出
305
各照会の選択ごとに
306
完了
(図５)
リポジトリ抽象化から照会フィールド定義を取得
308
各前回の照会ごとに
310
完了
実行
フィールドは使用されたか？
312
NO
YES
前回の条件に合わせてこの条件の分析を実行
314
アクションは必要か？
316
NO
YES
アクションを実行
318
致命的？
320
NO
YES
照会を停止
322
フィールド用の具体照会寄与部を構築
324
具体照会ステートメントに付加
321

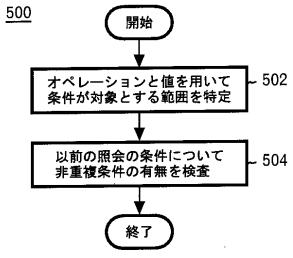
【図５】


300
照会後分析
336
非重複
照会結合分析
削減分析
338
340
342
照会を実行
322
NO
306
完了
結果フィールドはまだあるか？
328
YES
リポジトリ抽象化からフィールド定義を取得
330
フィールド用の具体照会寄与部を構築
332
具体照会ステートメントに付加
334

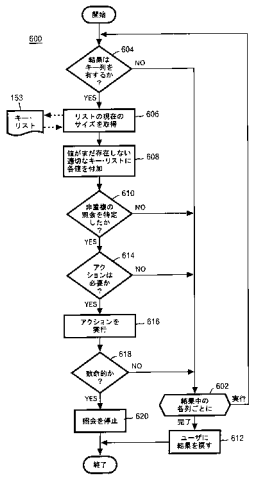
【図６】


フィールド用の具体照会寄与部を構築
単純なアクセス方式か？
402
NO
YES
フィールドの物理的な所在に基づいて照会寄与部を構築
404
継続
フィルタリング済みアクセス方式か？
406
NO
YES
フィールドの物理的な所在に基づいて照会寄与部を構築
408
フィルタの選択で寄与部を拡張
410
継続
合成アクセス方式か？
412
NO
YES
合成式のフィールドの物理的所在を検索・取得
414
物理的所在と合成式を用いて寄与部を作成
416
継続
他のアクセス方式の処理
418
400

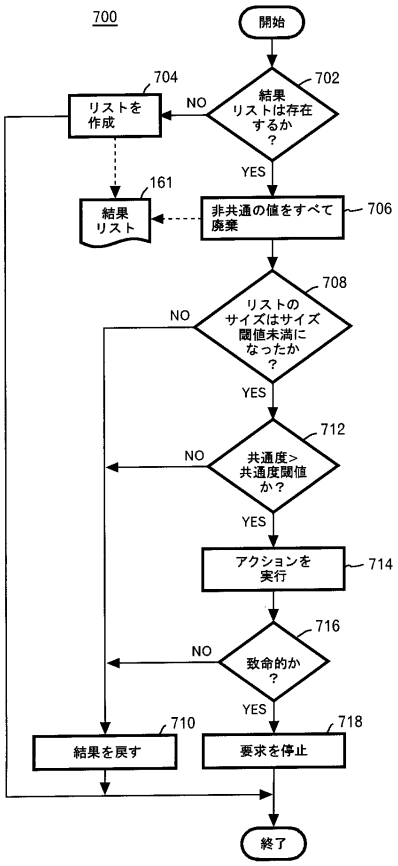
【図７】


500
開始
オペレーションと値を用いて条件が対象とする範囲を特定
502
以前の照会の条件について非重複条件の有無を検査
504
終了

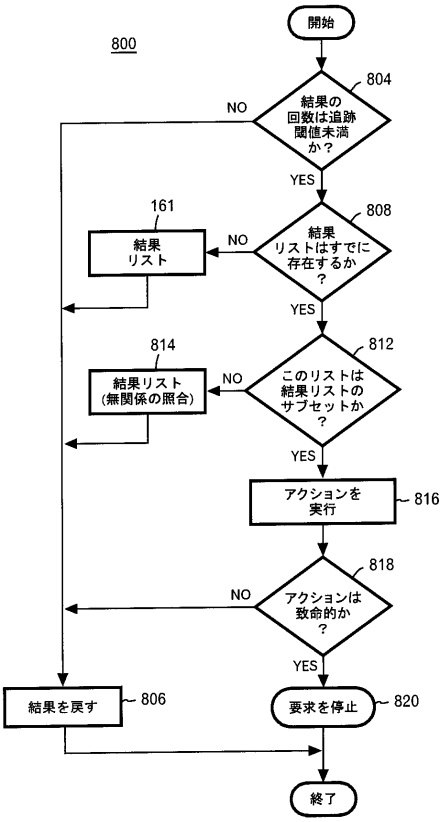
【図８】


開始
600
604
NO
YES
153
キーリスト
606
リストの現在のサイズを取得
608
610
NO
YES
614
NO
YES
616
アクションを実行
618
NO
YES
602
実行
620
照会を停止
完了
612
ユーザに結果を戻す
終了

【図９】


700
開始
704
リストを作成
NO
702
結果リストは存在するか？
YES
161
結果リスト
非共通の値をすべて廃棄
706
708
NO
リストのサイズはサイズ閾値未満になったか？
YES
712
NO
共通度>共通度閾値か？
YES
アクションを実行
714
716
NO
致命的か？
YES
710
結果を戻す
718
要求を停止
終了

【図１０】


開始
800
804
NO
結果の回数は追跡閾値未満か？
YES
161
結果リスト
808
NO
結果リストはすでに存在するか？
YES
814
結果リスト（無関係の照合）
812
NO
このリストは結果リストのサブセットか？
YES
アクションを実行
816
818
NO
アクションは致命的か？
YES
結果を戻す
806
要求を停止
820
終了

フロントページの続き

(72)発明者　デッティンジャー、リチャード、ディー
アメリカ合衆国　５５９０１　ミネソタ州　ロチェスター、ケンジントン　レーン　ノースウエスト　５３０５
(72)発明者　スティーヴンズ、リチャード、ジェイ
アメリカ合衆国　５５９５５　ミネソタ州　マントーヴィル、２５２エヌ・ディー　アベニュー　６１４３２

審査官　高橋　克

(56)参考文献　特開平１０－０４０２６６（ＪＰ，Ａ）
Nabil R. Adam and John C. Worthmann, Security-Control Methods for Statistical Databases: A Comparative Study, ACM Computing Surveys , １９８９年１２月, Vol. 21, No. 4, pp. 515-556
Raymond W. Yip and Karl N. Levitt, Data Level Inference Detection in Database Systems , Proceedings of the 11th IEEE workshop on Computer Security Foundations , IEEE Computer Society, １９９８年　6月　9日, pp.179-189

(58)調査した分野(Int.Cl., ＤＢ名)
G06F 21/24
G06F 12/00
G06F 17/30