(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2017-120645
(P2017-120645A)
(43)公開日　平成29年7月6日(2017.7.6)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 17/30	(2006.01)	G06F	17/30	330B	
		G06F	17/30	240A	
		G06F	17/30	310A	

審査請求 有　請求項の数 7　ＯＬ　（全 21 頁）

(21)出願番号　特願2016-254984 (P2016-254984)
(22)出願日　平成28年12月28日 (2016.12.28)
(31)優先権主張番号　10-2015-0187712
(32)優先日　平成27年12月28日 (2015.12.28)
(33)優先権主張国　韓国 (KR)

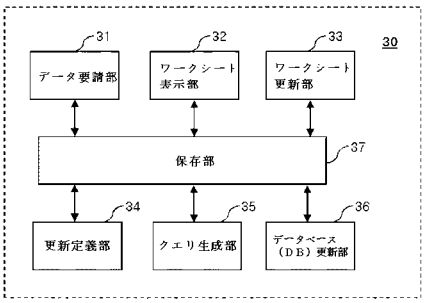
（特許庁注：以下のものは登録商標）
1．ＪＡＶＡ
2．ＶＩＳＵＡＬ　ＢＡＳＩＣ

(71)出願人　512096861
株式会社　ビーアイマトリックス
ＢＩ　ＭＡＴＲＩＸ　ＣＯ．，ＬＴＤ
大韓民国ソウル市江南区宣陵路４３３（駅三洞、新館１７階）
Ｎｅｗ　ｂｕｉｌｄｉｎｇ　１７Ｆ．，４３３，Ｓｅｏｌｌｅｕｎｇ－ｒｏ，Ｇａｎｇｎａｍ－ｇｕ，Ｓｅｏｕｌ，Ｒｅｐｕｂｌｉｃ　ｏｆ　Ｋｏｒｅａ
(74)代理人　110001427
特許業務法人前田特許事務所
(72)発明者　ベ　ヨングン
大韓民国　ソウル，カンナムグ，クァンピョンロ　１０ギル，６，１０２－３０１

最終頁に続く

(54)【発明の名称】クエリテンプレートを用いた、スプレッドシートインタフェースに基づくデータベースの一括更新システム

(57)【要約】　（修正有）
【課題】容易に更新が行えるデータベース一括更新システムを提供する。
【解決手段】更新サーバ３０は、スプレッドシートインタフェースを提供し、ワークシートを生成してデータセットの構造及びレコードデータをワークシートの上に表示し、スプレッドシートインタフェースを用いたユーザの入力に応じてレコードデータを更新し、各レコードデータの更新状態を表示するワークシート更新部と、ワークシート上で更新されたレコードに対して、更新されたレコードのデータセットの構造及びレコードデータを定義する更新定義部と、定義されたレコードのデータセットの構造を用いてクエリテンプレートを生成しておき、更新されたレコードのそれぞれに対してクエリテンプレートのマッピング変数に当該レコードデータをマッピングして実際のクエリ文を生成するクエリ生成部と、生成されたクエリ文をＤＢに転送して対象テーブルを更新するデータベース更新部と、を備える。
【選択図】図３



【特許請求の範囲】
【請求項１】
データベースとネットワークにより結ばれる、スプレッドシートインタフェースに基づくデータベースの一括更新システムにおいて、
前記データベースから対象テーブルのデータセットを要請して取り出すデータ要請部と、
スプレッドシートインタフェースを提供し、ワークシートを生成して前記データセットの構造及びレコードデータを前記ワークシートの上に表示するワークシート表示部と、
前記スプレッドシートインタフェースを用いたユーザの入力に応じて前記ワークシート上のレコードデータを更新し、各レコードデータの更新状態を表示するワークシート更新部と、
前記ワークシート上で更新されたレコードに対して、更新されたレコードのデータセットの構造及びレコードデータを定義する更新定義部と、
定義されたレコードのデータセットの構造を用いてクエリテンプレートを生成しておき、更新されたレコードのそれぞれに対して前記クエリテンプレートのマッピング変数に当該レコードデータをマッピングして実際のクエリ文を生成するクエリ生成部と、
生成されたクエリ文を前記データベースに転送して前記対象テーブルを更新するデータベース（ＤＢ）更新部と、
を備えることを特徴とするスプレッドシートインタフェースに基づくデータベースの一括更新システム。
【請求項２】
前記クエリ生成部は、前記更新されたレコードをストリームとして受信し、ストリームとして受信された順番に、各更新すべきレコードの内容をクエリ文としてリアルタイムにて生成することを特徴とする請求項１に記載のスプレッドシートインタフェースに基づくデータベースの一括更新システム。
【請求項３】
前記データベース（ＤＢ）更新部は、多数のレコード別のクエリ文をバッチでくくって一括処理するように前記データベースに要請することを特徴とする請求項１に記載のスプレッドシートインタフェースに基づくデータベースの一括更新システム。
【請求項４】
前記ワークシート表示部は、前記データセットの構造を前記ワークシート上のいずれか１行に表示し、データセットの構造が表示された行の下の行から前記レコードのデータを表示することを特徴とする請求項１に記載のスプレッドシートインタフェースに基づくデータベースの一括更新システム。
【請求項５】
前記ワークシート表示部は、前記ワークシートの上にレコードの更新状態を表示する第１の更新状態フィールドを更に追加してレコードフィールドを生成し、
前記ワークシート更新部は、前記ワークシート上の一つのレコードのセル内容が更新されると、更新の種類を当該レコードの第１の更新状態フィールドに表示し、
前記更新定義部は、前記ワークシート上の第１の更新状態フィールドを参照して、各レコードに対して更新有無及び更新の種類を判断することを特徴とする請求項１に記載のスプレッドシートインタフェースに基づくデータベースの一括更新システム。
【請求項６】
前記ワークシート表示部は、前記ワークシートを生成するとき、前記ワークシートと連動する実行モジュールに前記ワークシート上のレコード内容が更新されるとイベントが発生して、イベントにより呼び出されて実行されるスクリプト関数を含め、前記スクリプト関数は、当該レコードが更新されると、更新の種類を前記第１の更新状態フィールドに自動的に表示することを特徴とする請求項５に記載のスプレッドシートインタフェースに基づくデータベースの一括更新システム。
【請求項７】

前記ワークシート表示部は、前記ワークシートの上にフィールドの更新状態を表示する第２の更新状態フィールドを構成するレコードを更に追加して生成し、
前記ワークシート更新部は、前記ワークシート上の一つのレコードのセル内容が変更の更新が行われると、変更の更新の種類を当該レコードフィールドの第２の更新状態フィールドに表示し、
前記更新定義部は、レコードフィールドのうちキーフィールド及び第２の更新状態フィールドに変更の更新が行われたフィールドのみを構成して更新すべきデータセットの構造を生成することを特徴とする請求項５に記載のスプレッドシートインタフェースに基づくデータベースの一括更新システム。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、データベースのデータセットをエクセルなどスプレッドシートのワークシートとして取り出して修正し、修正されたレコード別にクエリ文を作成してデータベースを更新するが、クエリテンプレートを予め生成し、更新されたデータ値のみを置き換えて実際のクエリ文を生成する、スプレッドシートインタフェースに基づくデータベースの一括更新システムに関する。

【背景技術】

【０００２】

一般に、ウェブ上でデータベースを更新するためには、データベースを処理するウェブプログラム及びこれを指示するためのクエリ文が必要である。すなわち、データベース管理システムにおいて構造化問い合わせ言語（ＳＱＬ）のレベルにより更新演算が行われる（例えば、下記の特許文献１参照）。

【０００３】

具体的に、データベースに特定のテーブルに書き込まれたデータを入力したり、特定のテーブルに書き込まれたデータを修正したり、データベースから特定のテーブルに書き込まれたデータを削除したりするためには、ジャバサーバページス（ＪＳＰ：Ｊａｖａ Ｓｅｒｖｅｒ Ｐａｇｅｓ）、アクティブサーバページス（ＡＳＰ：Ａｃｔｉｖｅ Ｓｅｒｖｅｒ Ｐａｇｅｓ）、ハイパテキストプリプロセッサ（ＰＨＰ：Ｈｙｐｅｒｔｅｘｔ Ｐｒｅｐｒｏｃｅｓｓｏｒ）などサーバ側のウェブプログラムを用いて更新を行う。具体的に、前記サーバ側のウェブプログラムはデータベースと接続し、対象テーブルのデータを処理するための構造化問い合わせ言語（ＳＱＬ）文などクエリ文を作成せねばならない。作成されたクエリ文をデータベースに転送し、当該クエリを処理するように要請すると、データベースは受信した当該クエリを処理することにより、データベース内のテーブルが更新される。

【０００４】

このため、上述したウェブプログラムがデータベースのクエリ文を処理するために、それぞれのテーブル又は当該項目別のウェブ画面を構成せねばならない。また、対象テーブルの数に見合う分だけ入力画面を別途に開発せねばならないという問題がある。この理由から、それぞれのクエリ文も開発者が自ら作成せねばならないため、多大な時間及び工数がかかるという問題がある。なお、作成された画面でも単件データ又は少量のデータしか取り扱うことができず、大量のデータの処理には不便さがあり、しかも、処理に長い時間がかかる。

【０００５】

このために、データベースをより効率よく修正するために、ネットワークに基づいて遠隔データベースをローカルで修正する様々な技術が提案されている。例えば、ユーザ装置上で多次元データベースの一部をキャッシングして修正して、これを多次元データベースに反映して、データベースを修正する技術が提案されている（例えば、下記の特許文献２参照）。また、周期的にサンドファイルなど特定のファイルフォーマットを用いて更新する技術が提案されている（例えば、下記の特許文献３参照）。更に、部分データベースに

おいてサブセットデータベースの更新を処理する技術が提案されている（例えば、下記の特許文献４参照）。
【０００６】
しかしながら、下記の特許文献に記載の前記従来の技術は、いずれもデータベースの一部である他のデータベースを修正して、これを母データベースに反映する技術である。このため、ローカルでもデータベースを構成して処理せねばならないという問題がある。すなわち、ローカルユーザは、データベースを取り扱う知識を有さなければならないという問題がある。
【０００７】
また、大量のデータをデータベースに取り込むための技術も提案されている（例えば、下記の特許文献５参照）。すなわち、データの集まりを変換データに変換する多数の連続した作業を含む処理チェーンを実行って、変換データを基準としてデータベースを変更する技術である。しかしながら、下記の特許文献に記載の前記従来の技術は、新たに追加されるデータを特定のフォーマットに設定した後に、これらのデータを元のデータベースに変換して保存する技術である。このため、一般のユーザが大量のデータを特定のフォーマットに構成せねばならず、これを直観的に処理することができないという問題がある。
【先行技術文献】
【特許文献】
【０００８】
【特許文献１】大韓民国公開特許第１０－１９９９－００３８６９８号（１９９９年０６月０５日付け公開）
【特許文献２】大韓民国公開特許第１０－２００８－００３１２０２号（２００８年０４月０８日付け公開）
【特許文献３】大韓民国公開特許第１０－２００４－００５３２５４号（２００４年０６月２３日付け公開）
【特許文献４】大韓民国公開特許第１０－２０１２－００５２３０１号（２０１２年０５月２３日付け公開）
【特許文献５】大韓民国公開特許第１０－２０１４－００２７１２８号（２０１４年０３月０６日付け公開）
【発明の概要】
【発明が解決しようとする課題】
【０００９】
本発明は上記事情に鑑みてなされたものであり、その目的は、データベースのデータセットをスプレッドシートのワークシートとして取り出して修正した後、修正されたデータセットのレコードをデータベースに更新するが、データセットのレコードのフィールド構造を用いて更新テンプレートを生成した後、各レコード別のデータ値のみを置き換えて実際のクエリ文を生成して更新する、スプレッドシートインタフェースに基づくデータベースの一括更新システムを提供することである。
【００１０】
また、本発明の他の目的は、クエリ文はテンプレートとして一回のみ生成し、転送データから当該クエリの変数に当該カラム値のみをマッピングしてバッチに追加した後、最後にもう一回データベースに実行を要請する、スプレッドシートインタフェースに基づくデータベースの一括更新システムを提供することである。
【課題を解決するための手段】
【００１１】
前記目的を達成するために、本発明は、スプレッドシートインタフェースに基づくデータベースの一括更新システムに関するものであり、前記データベースから対象テーブルのデータセットを要請して取り出すデータ要請部と、スプレッドシートインタフェースを提供し、ワークシートを生成して前記データセットの構造及びレコードデータを前記ワークシートの上に表示するワークシート表示部と、前記スプレッドシートインタフェースを用

いたユーザの入力に応じて前記ワークシート上のレコードデータを更新し、各レコードデータの更新状態を表示するワークシート更新部と、前記ワークシート上で更新されたレコードに対して、更新されたレコードのデータセットの構造及びレコードデータを定義する更新定義部と、定義されたレコードのデータセットの構造を用いてクエリテンプレートを生成しておき、更新されたレコードのそれぞれに対して前記クエリテンプレートのマッピング変数に当該レコードデータをマッピングして実際のクエリ文を生成するクエリ生成部と、生成されたクエリ文を前記データベースに転送して前記対象テーブルを更新するデータベース（ＤＢ）更新部と、を備えることを特徴とする。

【００１２】

また、本発明は、スプレッドシートインタフェースに基づくデータベースの一括更新システムにおいて、前記クエリ生成部は、前記更新されたレコードをストリームとして受信し、ストリームとして受信された順番に、各更新すべきレコードの内容をクエリ文としてリアルタイムにて生成することが好ましい。

【００１３】

更に、本発明は、スプレッドシートインタフェースに基づくデータベースの一括更新システムにおいて、前記データベース（ＤＢ）更新部は、多数のレコード別のクエリ文をバッチでくくって一括処理するように前記データベースに要請することが好ましい。

【００１４】

更にまた、本発明は、スプレッドシートインタフェースに基づくデータベースの一括更新システムにおいて、前記ワークシート表示部は、前記データセットの構造を前記ワークシート上のいずれか１行に表示し、データセットの構造が表示された行の下の行から前記レコードのデータを表示することが好ましい。

【００１５】

更にまた、本発明は、スプレッドシートインタフェースに基づくデータベースの一括更新システムにおいて、前記ワークシート表示部は、前記ワークシートの上にレコードの更新状態を表示する第１の更新状態フィールドを更に追加してレコードフィールドを生成し、前記ワークシート更新部は、前記ワークシート上の一つのレコードのセル内容が更新されると、更新の種類を当該レコードの第１の更新状態フィールドに表示し、前記更新定義部は、前記ワークシート上の第１の更新状態フィールドを参照して、各レコードに対して更新有無及び更新の種類を判断することが好ましい。

【００１６】

更にまた、本発明は、スプレッドシートインタフェースに基づくデータベースの一括更新システムにおいて、前記ワークシート表示部は、前記ワークシートを生成するとき、前記ワークシートと連動する実行モジュールに前記ワークシート上のレコード内容が更新されるとイベントが発生して、イベントにより呼び出されて実行されるスクリプト関数を含め、前記スクリプト関数は、当該レコードが更新されると、更新の種類を前記第１の更新状態フィールドに自動的に表示することが好ましい。

【００１７】

更にまた、本発明は、スプレッドシートインタフェースに基づくデータベースの一括更新システムにおいて、前記ワークシート表示部は、前記ワークシートの上にフィールドの更新状態を表示する第２の更新状態フィールドを構成するレコードを更に追加して生成し、前記ワークシート更新部は、前記ワークシート上の一つのレコードのセル内容が変更の更新が行われると、変更の更新の種類を当該レコードフィールドの第２の更新状態フィールドに表示し、前記更新定義部は、レコードフィールドのうちキーフィールド及び第２の更新状態フィールドに変更の更新が行われたフィールドのみを構成して更新すべきデータセットの構造を生成することが好ましい。

【発明の効果】

【００１８】

以上述べたように、本発明によるスプレッドシートインタフェースに基づくデータベースの一括更新システムによれば、更新すべきクエリテンプレートを生成した後、各レコー

ド別のデータ値のみを置き換えて実際のクエリ文を生成することにより、更新されたデータをストリームとして受信してリアルタイムにてクエリ文を生成することができ、その結果、早い時間内にクエリ文を生成して処理することができるという効果が得られる。

【００１９】

また、本発明によるスプレッドシートインタフェースに基づくデータベースの一括更新システムによれば、クエリ文をテンプレートとして一回のみ生成し、転送データから当該クエリの変数に当該カラム値のみをマッピングしてバッチに追加した後、最後にもう一回データベースに実行を要請することにより、速やかに大量のデータを処理することができるという効果が得られる。

【００２０】

更に、本発明によるスプレッドシートインタフェースに基づくデータベースの一括更新システムによれば、事務用に多用されるエクセルなどスプレッドシートを用いてデータベースを更新することにより、ユーザはクエリ文などデータベースの処理方法に関する知識なしにも直観的にデータを更新してデータベースを更新することができるという効果が得られる。

【図面の簡単な説明】

【００２１】

【図１】本発明によるスプレッドシートインタフェースに基づくデータベースの一括更新システムを行うための全体システムの構成に対するブロック図である。

【図２】本発明によるスプレッドシートインタフェースに基づくデータベースの一括更新システムを行うための全体システムに対する構成図である。

【図３】本発明の一実施形態によるスプレッドシートインタフェースに基づくデータベースの一括更新システムの構成に対するブロック図である。

【図４】本発明の一実施形態による目的データベースの対象テーブルを選定する画面の一例である。

【図５】本発明の一実施形態によるワークシートの上にテーブルのスキーマを表示した画面の一例である。

【図６】本発明の一実施形態によるワークシートの上にテーブルのデータを表示した画面の一例である。

【図７】本発明の第１の実施形態によるワークシートの上にテーブルのデータが更新された状態を表示した画面の一例である。

【図８】本発明の第１の実施形態によりＸＭＬで表示されたテーブル情報の一例である。

【図９】本発明の第１の実施形態により更新すべきデータを定義したフラットファイルの一例である。

【図１０】本発明の第１の実施形態によるクエリテンプレートの一例である。

【図１１】本発明の第２の実施形態によるワークシートの上にテーブルのデータが更新された状態を表示した画面の一例である。

【図１２】本発明の第２の実施形態によりＸＭＬで表示されたテーブル情報の一例である。

【図１３】本発明の第２の実施形態により更新すべきデータを定義したフラットファイルの一例である。

【図１４】本発明の第２の実施形態によるクエリテンプレートの一例である。

【図１５】本発明の第２の実施形態により作成された実際のクエリに対する例示図である。

【発明を実施するための形態】

【００２２】

以下、本発明の実施のための具体的な内容について添付図面に基づいて説明する。

【００２３】

また、本発明を説明するに当たって、同じ部分には同じ符号を付し、その重複する説明は省略する。

【００２４】
まず、本発明によるスプレッドシートインタフェースに基づくデータベースの一括更新システムを行うための全体システムについて図１を参照して説明する。
【００２５】
図１に示すように、本発明を行うための全体システムは、ユーザ端末１０に設けられるクライアント２０と、クライアント２０とネットワークにより結ばれる更新サーバ３０と、データを保存して管理するデータベース４０と、を備える。図２は、同全体システムを示す構成図である。図２の記載の「ユーザ」がユーザ端末１０内の構成であり、クライアント２０により駆動される構成を表示している。
【００２６】
クライアント２０は、ユーザ端末１０に設けられるクライアント用プログラムシステムであり、ウェブブラウザを介してユーザインタフェースを有する。ウェブブラウザを介してユーザインタフェースを提供する。すなわち、ユーザは、ウェブブラウザ又はウェブブラウザなどの画面のインタフェースを介して、オンライン上でデータ処理作業を行う。このとき、ユーザ端末１０は、ユーザの指令などを入力されて当該指令を行い、処理結果を画面の上に又はウェブブラウザの上に表示する。
【００２７】
一方、ユーザ端末１０は、パソコン（ＰＣ）、ノート型パソコン、タブレットＰＣ、ファブレット、ＰＤＡ、スマートフォンなどコンピューティング機能を有するコンピュータ端末である。ユーザ端末１０及び更新サーバ３０はネットワークにより結ばれて、ユーザ端末１０上のクライアント２０はオンライン上でデータ処理作業を行う。
【００２８】
また、クライアント２０は、データ要請、データ更新などオンライン上で処理する作業を更新サーバ３０に要請し、その結果をサーバ３０から取り出してウェブブラウザの上に表示する。或いは、処理されたデータを更新サーバ３０に転送する。特に、クライアント２０は、データベース４０から取り出されたデータをエクセル（マイクロソフト社製のスプレッドシートソフトウェア商標である）などスプレッドシートプログラムのインタフェースを介して表示するか、或いは、指令／データを入力される。なお、クライアント２０は、スプレッドシートインタフェースを介して処理されたデータの更新結果を更新サーバ３０に転送する。
【００２９】
次いで、更新サーバ３０は、データベース４０からデータセットを取り出してクライアント２０に転送するか、或いは、クライアント２０から更新されたデータセットを回答してデータベース４０を更新する。特に、更新サーバ３０は、クライアント２０から新たに追加されたり、削除されたり、修正されたりしたデータを検出して、変更されたデータを処理するためのデータベースクエリ文を自動的に作成する。更新サーバ３０は、作成されたクエリ文をデータベース４０に要請して、その結果がデータベース４０に反映されるように要請する。
【００３０】
一方、上述したクライアント２０及び更新サーバ３０の機能は一例に過ぎず、サーバ及びクライアントの実現技術に応じて種々に具体化される。すなわち、クライアント２０及び更新サーバ３０は、一つの更新システムであり、これらの機能が性能に応じて互いに分配される。
【００３１】
他の例によれば、クライアント２０は、単にウェブブラウザ機能のみを有しており、全ての機能は更新サーバ３０において構築される。すなわち、更新サーバ３０は、データベース４０のインタフェース機能及び修正されたデータに対するクエリ文の作成機能だけではなく、スプレッドシートに基づくインタフェースを用いてデータを修正する機能をいずれも有する。
【００３２】

更に他の例によれば、更新サーバ３０は、単にデータベース４０とのインタフェース機能のみを有しており、クライアント２０は、スプレッドシートに基づくインタフェースを介してデータの修正機能だけではなく、修正されたデータに対するクエリ文の自動生成機能まで有する。

【００３３】

更に他の例によれば、更新サーバ３０なしに、クライアント２０が直ちにデータベース４０に接続して、上述したデータベース接続機能、スプレッドシートインタフェースを用いたデータの修正機能及びクエリ文の自動生成機能などをいずれも行う。

【００３４】

次いで、データベース６０は、データを保存するための通常のデータベース（ＤＢ）であり、データを管理するためのデータベース管理システム（ＤＢＭＳ）を備え、データの保存、削除、検索などの作業をクエリを用いて行う。特に、データベース４０は、商用化されたデータベースであり、データセットを処理するための通常のクエリ機能を用いてデータクエリサービスを行う。

【００３５】

特に、データベース４０は、ビッグデータを保存するデータベースである。また、好ましくは、データベース４０は、関係型データベース（ＲＤＢ）である。

【００３６】

次いで、本発明の第１の実施形態によるスプレッドシートインタフェースに基づくデータベースの一括更新システムについて図３を参照してより具体的に説明する。上述したように、データベース自動更新システムは、ユーザ端末１０のクライアント２０及びネットワークに接続されたサーバ３０を備える実現されるサーバークライアントシステムである。しかしながら、以下、サーバ３０において実現されるシステムを例にとって説明する。

【００３７】

図３に示すように、本発明の一実施形態によるスプレッドシートインタフェースに基づくデータベースの一括更新システムは、データベースのデータセットを要請するデータ要請部３１と、要請したデータセットをスプレッドシートインタフェースに表示するワークシート表示部３２と、ワークシート上でデータセットのうちの一部のデータを更新するワークシート更新部３３と、更新されたデータに対して更新内容を定義する更新定義部３４と、更新されたデータを更新するためのクエリ文を生成するクエリ生成部３５と、生成されたクエリ文を用いてデータベースを更新するデータベース（ＤＢ）更新部３６と、を備える。

【００３８】

まず、データ要請部３１は、データベース４０に要請して目的データベースからデータセットを取り出す。ここで、データセットとは、同じフィールドにより構成される多数のレコードにより構成されるデータの集まりのことをいう。すなわち、一つのレコードは、多数のフィールド値により構成されるが、全てのレコードのフィールド（カラム）は、同じフィールド（又は、カラム）を有する。

【００３９】

好ましくは、データセットは、一つのデータベース（ＤＢ）テーブルから取り出されたデータである。

【００４０】

具体的に、データ要請部３１は、データベース４０に要請して、目的データベースのテーブル一覧を問い合わせ、問い合わせられたテーブル一覧を画面に表示する。ユーザは、画面に表示されたテーブル一覧のうち所望のテーブルを選択する。図４は、テーブル一覧のうち取り出すテーブルを選択する画面の一例を示す。

【００４１】

ユーザ画面に表示されたテーブルのうちから一つが選択されると、当該テーブルに関するスキーマ情報を取り出す。目的データベース又は対象テーブルのスキーマは、テーブルのデータがどのように構成されているかに関する細部情報である。すなわち、スキーマは

、各データフィールドの名前、表示名、データのタイプ（種類）、条件などをテーブルデータの構造を示す。

【００４２】

他の例によれば、データ要請部３１は、目的データベースのテーブルデータをクエリ文を介して取り出す。好ましくは、このときのクエリ文は、一つのテーブルに対するクエリ文であり、必ずクエリ結果内に当該テーブルのキーフィールドが存在しなければならない。このとき、多数のフィールドがキーを形成する場合、当該多数のフィールドがいずれも含まれなければならない。クエリ文を介して取り出されたデータは、一つのデータセットを形成する。

【００４３】

すなわち、データ要請部３１は、目的データベースからデータセットを取り出す。 前記テーブルを選択する場合には、取り出すデータセットは当該テーブルの全てのデータであり、対象テーブルからクエリ文を取り出した場合には、クエリ文によるフィルタリングされたテーブルの一部である。このとき、好ましくは、ジャバデータベース（ＤＢ）接続ドライバ（ＪＤＢＣ）などを介して目的データベースのデータを取り出す。 前記データスキーマもジャバデータベース（ＤＢ）接続ドライバ（ＪＤＢＣ）を用いて取り出す。

【００４４】

一方、他の実施形態によれば、データ要請部３１は、目的データベースの対象テーブルをデータセットとして取り出し、取り出されたデータセット（又は、対象テーブル）に対してクエリを行うクエリ編集器を用いて、クエリ結果を抽出する。また、抽出されたクエリ結果を一つのデータセットとして用いる。すなわち、後述するワークシート表示部３２から抽出されたクエリ結果のデータセットのみをワークシートの上にデータとして表示する。これにより、対象テーブルの全体をワークシートの上に表示せずに、ユーザが希望するフィールドのみにより構成されたテーブルをワークシートの上に表示することができる。なお、不要なデータフィールドを表示しないことから、ユーザのデータ編集の使い勝手がよくなる。

【００４５】

上述したように、目的データベースから取り出されたテーブルのデータセット、又はクエリ文により取り出されたデータセット、又はテーブルをクエリ文によりフィルタリングしたデータセットなどはいずれも更新すべきデータセットである。以下、これを初期のデータセット（第１のデータセット）と称する。

【００４６】

次いで、ワークシート表示部３２は、スプレッドシートインタフェースを提供し、エクセルなどスプレッドシートの上にワークシート（以下、ワークシートと称する。）を生成し、生成されたワークシートの上にデータセットを表示する。特に、目的データベースから取り出されたデータセットをスプレッドシートのワークシートの上に表示する。スプレッドシートは、エクセル（マイクロソフト社製のプログラム商標）、ロータス（Ｌｏｔｕｓ １－２－３、ロータスソフトウェア社製のプログラム商標）、ハンセル（ハングル及びコンピュータ社製のプログラム商標）など、行及び列により区分された表の形式で表示されるワークシートにより構成されて、各セルにデータを処理するスプレッドシートソフトウェアプログラムである。

【００４７】

まず、ワークシート表示部３２は、対象データセットのスキーマ情報（又は、データセットの構造）を分析して各フィールドに関する詳細情報に合わせてワークシートに作成する。このために、ワークシートを新たに生成する。

【００４８】

その一例として、図５に示すように、ワークシートの列に応じて、各データフィールドの名前、表示名、データのタイプ（種類）、条件、キーフィールドなどテーブルデータの細部構造を表示する。また、各列に相当するデータのタイプを解析して、各列の表示属性（特に、数字に対する表示形式）をデータのタイプ（形式）に合わせて補正する。例えば

、列の名前が「誕生日」であるが、当該フィールドの形式が日付け形式であれば、当該列を全て日付け形式に設定する。なお、通話表示であれば、これによるデータ形式で表示する。

【００４９】

また、ワークシート表示部３２は、スプレッドシートのワークシートを生成するとき、スプレッドシートから提供するワークシートに連動される実行モジュールを一緒に生成する。実行モジュールは、ＶＢＡ（Ｖｉｓｕａｌ Ｂａｓｉｃ ｆｏｒ ａｐｐｌｉｃａｔｉｏｎ）などスクリプト方式のプログラム言語により作成されたモジュール（又は、コード）である。実行モジュールは、ワークシート内において特定のイベントが発生すると、各イベントに対応する関数が実行されるようにするモジュールである。このため、ワークシート更新部３３によりワークシート上でユーザが削除、修正、挿入など特定の作業を行うと、当該作業又は作業イベントに応じて前記実行モジュール内の関数が実行されて、関連作業を行う。

【００５０】

更に、ワークシート表示部３２は、更新すべきデータセット（又は、初期のデータセット）をスプレッドシートのワークシートに表示する。図６に示すように、前記データの各フィールドのフィールド名など細部構造が前記ワークシートの上に列をなして平行に記載されており、当該フィールド名に対応する列内の下の行に連続して当該フィールド値（又は、テーブルデータ）を表示する。すなわち、図６に示すように、スプレッドシート（又は、ワークシート）上のデータは、行及び列により各セルにデータが表示され、このとき、上段行内における各列にはフィールド名（フィールド名前）、表示名、データタイプなどが記載され、その下にレコードを各フィールドに合わせて配列する。

【００５１】

すなわち、初期のデータセットは、多数のレコードにより構成され、各レコードは、ワークシート上で一つの行に表示される。レコードの各列に相当するセルには当該レコードフィールドのフィールド値が表示される。特に、前記各フィールドの列に対するデータ形式を分析して、当該形式に全ての列を設定したため、当該レコードの当該列はいずれもそのデータ形式に合うようにスプレッドシートのワークシートの上に表示される。

【００５２】

図６の例において、ＥＭＰＬＯＹＥＥＳテーブルの場合、ＥＭＰＬＯＹＥＥ＿ＩＤが基本キーであるため、このカラム（又は、列）は重複を許容しない。

【００５３】

また、ワークシート表示部３２は、一つのワークシートの上に表示可能なレコードの最大数を予め設定する。そして、当該最大数を超えると、新たなワークシートを生成し、残りのレコードを生成されたワークシートに表示する。例えば、スプレッドシート画面においてデータを処理するため、同じシート内において１００万件、複数のシートを介して数百万件のデータを定義し、データベースに入力することができる。

【００５４】

一方、ワークシート表示部３２は、エクセルなど商用化されたスプレッドシートプログラムの主な機能のみを埋め込んで用いる。すなわち、スプレッドシート処理作業は、スプレッドシートのコア機能により行われ、全体のインタフェースは、ワークシート表示部３２により処理される。後述するワークシート更新部３３も同様である。

【００５５】

更に、ワークシート表示部３２は、更新状態を表示する列を更に表示する。以下、当該列を更新状態フィールド又は更新状態列と称する。好ましくは、ワークシート上で最初の列を更新状態フィールドに設定する。図５の例において、「Ｃｈｅｃｋ」と記載されている列が更新状態フィールドに設定されている。

【００５６】

更にまた、ワークシート表示部３２は、指令を行うための指令ボタンをワークシートの上に表示する。好ましくは、指令ボタンは、テーブル又はクエリ結果を問い合わせるため

の問い合わせボタンや、更新されたデータをサーバに転送してデータベースを更新するための確認ボタンなどを備える。

【００５７】

次いで、ワークシート更新部３３は、ワークシート上に表示されたデータに対して更新作業を行う。特に、ワークシート更新部３３は、前記ワークシート表示部３２が提供するスプレッドシートインタフェース画面を介して、ユーザがデータ更新作業を行う。

【００５８】

すなわち、ワークシート更新部３３は、初期のデータセットをスプレッドシートのワークシートの上に表示し、ユーザにより処理されるスプレッドシートのワークシート上の作業を処理する。例えば、ユーザがスプレッドシート上で特定のセルのデータを修正したり、特定の行を挿入したり、削除したりする。このとき、通常のスプレッドシートインタフェースのようなタイプにインタフェースを処理する。

【００５９】

また、ワークシート更新部３３の各機能は、ワークシートと連動される実行モジュール（ＶＢＡなどスクリプト言語により作成されたプログラムモジュール）により実現される。すなわち、ワークシート上でユーザが修正、挿入、削除など特定の作業を行うと、当該作業を一つのイベントと判断して、当該作業イベントに対応する実行モジュールの関数が実行される。このような機能は、スプレッドシートのコア機能から提供される。以下、説明の便宜のために、修正、削除、挿入（新規追加）などによりデータの内容が異なってきたものをいずれも「更新」されたものとする。

【００６０】

ワークシート更新部３３は、ワークシート上で編集作業が発生されると、編集種類又は更新の種類に応じて更新状態フィールドのセルに更新の種類を表示する。好ましくは、更新の種類は、記号又は更新の種類記号で表示される。更新の種類記号は、予め設定された記号である。更新の種類は、当該レコード又は当該行の更新状態又は更新の種類を表示するものであり、修正、挿入、削除などの各レコードに対する更新状態を表示する。例えば、データのセルが修正されると、当該行の更新状態カラム（列）に「Ｕ」が表示され、行が追加されると、「Ｃ」が表示される。

【００６１】

また、ワークシート更新部３３は、更新状態フィールドにユーザが自ら更新状態又は更新の種類記号を表示するために配設される。例えば、ユーザが削除する行（又は、レコード）の更新状態カラムに「Ｄ」を入力する。このとき、更新状態列（カラム）の各セルにはコンボボックスを設けて、各レコード別（又は、行別）に更新状態を自ら設定するようにする。

【００６２】

更に、好ましくは、いずれの項目も更新されていないレコード（又は、行）には更新状態フィールドにいかなる値又は記号も表示されない。

【００６３】

図７は、図６のレコードデータのうちの一部が更新された状態を示す。すなわち、キー１０４のレコードが削除され、レコード１０５、１０６、１０８がそれぞれ変更され、変更されたデータは給料及び電話番号ｐｈｏｎｅ＿ｎｕｍｂｅｒである。このため、更新フィールドｃｈｅｃｋ１にはそれぞれＤ、Ｕ、Ｕ、Ｕが表示される。

【００６４】

次いで、更新定義部３４は、ワークシート又はワークシート上で更新されたデータを検出して、検出されたデータ（又は、レコード）に対して更新内容を定義する。好ましくは、ＸＭＬドキュメントの形で定義し、データはフラットファイルにより構成する。

【００６５】

前記ワークシート更新部３３により、ユーザは、ワークシート上のデータセットをセル、範囲、行又は列単位で修正する。すなわち、一つのセル内にあるデータを修正したり、一つの行（又は、レコード）などを全て削除したりする（すなわち、削除すべきレコード

の更新状態フィールドに削除記号Ｄを表示する）。或いは、新たなデータレコードを挿入し、各フィールド内に値を入力する。

【００６６】

また、更新定義部３４は、検出された更新レコードに対してそれぞれ更新内容を定められた定義規則により定義する。このとき、定義すべき内容は、更新すべきデータ又はデータセットの構造又はスキーマを表示するテーブル情報及び更新すべきデータである。すなわち、更新定義部３４は、更新されたデータ又は更新されたレコードに対してのみ更新データを書き込む（又は、定義する）。すなわち、全く更新されていないデータは書き込まない。

【００６７】

まず、更新定義部３４は、データセット（又は、更新データ構造）を予め定められたフォーマット（又は、定義規則）により定義する。好ましくは、データセットの構造（又は、更新データ構造）をＸＭＬ文字で表示する。データセットの構造をＸＭＬドキュメントで表示した一例が図８に示されている。

【００６８】

図８に示すテーブル情報又はデータセットの情報は、下記のような個体及びその属性に関するデータを含む。好ましくは、テーブル情報又はデータセットの構造に関する情報は、テーブル名（ユーザ画面からテーブルを選択するときに自動的に指定される）、データベース（ＤＢ）接続コード（ユーザ画面においてデータベース（ＤＢ）に接続されるときに自動的に指定される）、キーフィールドの存否、データタイプ、ヌル（ＮＵＬＬ）値の許容有無などの内容を含む。

【００６９】

（１）ＥｘｅｃｕｔｅＩｎｆｏ：実行すべきデータベース情報

－ＣｏｎｎｅｃｔｉｏｎＣｏｄｅ：サーバに定義されたデータベースコード

（２）ＤａｔａＴａｂｌｅ：更新すべきテーブル情報、多数のテーブルを同じトランザクションで処理可能

－Ｎａｍｅ：テーブル名

－Ｃａｐｔｉｏｎ：表示名

－ＴａｒｇｅｔＴａｂｌｅ：更新すべきテーブル名

（３）Ｃｏｌｕｍｎ：テーブル（又は、データセット）のカラム情報

－Ｎａｍｅ：カラム名

－Ｃａｐｔｉｏｎ：カラム表示名

－Ｃｏｎｓｔｒａｉｎｔ：ｋｅｙ情報 Ｐ：Ｐｒｉｍａｒｙ ｋｅｙ Ｖ：ｎｏｔｎｕｌｌ Ｎ：ｎｕｌｌａｂｌｅ

－ＤａｔａＴｙｐｅ：データタイプ Ｃ：Ｓｔｒｉｎｇ Ｎ：数字型、Ｄ：Ｄａｔａ型

－ＵｐｄａｔｅＦｌａｇ：更新クエリの生成時にカラムを含むか否か

（４）Ｒｏｗｓ：処理すべきデータを定義する

－ＲｏｗＩｎｄｅｘＣｏｌ：データ行の通し番号が位置する列

－ＲｏｗＳｔａｔｅＣｏｌ：データの件別の処理に際して当該項を処理する区切りコードが位置する列

－ＴｅｘｔＴｙｐｅ：処理すべきデータの位置 ＦＩＬＥ：添付ファイルとして転送する ＸＭＬ：ｘｍｌを内部に含む

データセットのカラム情報又はフィールド情報は、対象テーブルの全体のフィールド情報を示してもよく、その一部を示してもよい。すなわち、前記対象テーブルをクエリを介して一部のみ取り出す場合には、テーブルのフィールドのうちの一部に対してのみフィールド構造を生成する。

【００７０】

また、更新定義部３４は、更新すべきデータを予め定められたフォーマットで書き込んでファイルとして生成する。好ましくは、更に、更新定義部３４は、ワークシート上で更新されたデータをフラットファイルとして生成する。フラットファイルとは、データを一

連の列により構成する通常のファイルのことをいう。

【００７１】

図９は、フラットファイルの一例を示すものである。好ましくは、スプレッドシート画面に定義されたデータは、サーバと予め定義されたカラム、行区切り子により図９に示す形式のテキストファイルとして作成されてサーバに転送される。ここで、例えば、カラム区切り子を「｜」で示し、行区切り子を「エンター（又は、改行）」記号で示す。

【００７２】

特に、図９において、各行の先頭に当該レコード（当該行）がどのように処理されるかに対する更新の種類を表示する。各更新状態の記号は、下記の通りである。

【００７３】

Ｃ：新規入力　Ｕ：データの更新　Ｄ：データの削除

次いで、クエリ生成部３５は、テーブル情報及び更新すべきデータが定義された更新ドキュメントを受信して、更新されたデータに対してクエリ文を生成する。このとき、クエリ生成部３５は、データセットの構造を用いてクエリテンプレートを予め生成しておき、更新すべきデータのそれぞれに対してクエリテンプレートにマッピングして実際に更新するクエリを生成する。

【００７４】

すなわち、クエリ生成部３５は、テーブル情報を定義した更新ドキュメントの内容を解析して、更新すべきデータベース、テーブル、カラム情報を抽出する。また、抽出された情報を用いて、入力クエリ、更新クエリ、削除クエリに対するクエリテンプレートなどを生成する。

【００７５】

クエリテンプレートの例が図１０に示されている。図１０は、図８のデータセットの構造により作成されたクエリテンプレートである。図１０（ａ）は、入力クエリ（又は、挿入クエリ）テンプレートであり、図１０（ｂ）は、更新クエリ（変更クエリ又は更新クエリ）のテンプレートである。なお、図１０（ｃ）は、削除クエリのクエリテンプレートを示す。

【００７６】

クエリ生成部３５は、入力、更新、削除など当該クエリ文のフォーマットを作成し、クエリ指令フォーマットにテーブル名及びフィールド名（カラム名）などを記載し、且つ、変数値（又は、カラム値）に相当する部分に変数値の代わりにマッピング変数を挿入して、クエリテンプレートを生成する。このとき、マッピング変数には順番に番号が付けられる。

【００７７】

図１０の例において、「ＩＮＳＥＲＴ　ＩＮＴＯ＜テーブル名＞（カラム名，．．．）ＶＡＬＵＥＳ（カラム値，．．．）」などがクエリ文のフォーマットであり、ＥＭＰＬＯＹＥＥＳがテーブル名であり、ＥＭＰＬＯＹＥＥ＿ＩＤ、ＦＩＲＳＴ＿ＮＡＭＥなどがカラム名である。また、特殊文字@を用いた@Ｄ１，@Ｄ２，．．．，@ＤＮなどがマッピング変数である。１，２，．．．，Ｎが番号の付けられた数字を示す。マッピング変数は、クエリ文のフォーマットにおけるカラム値又は変数値に挿入される部分に記載される。

【００７８】

マッピング変数に番号の付けられた数字は、前記マッピング変数にマッピングされるべき更新データの順序を示す。すなわち、更新データのｎ番目のデータ（フィールド値、カラム値）がｎ番目に番号の付けられたマッピング変数である。

【００７９】

次いで、クエリ生成部３５は、データをそれぞれレコード別に読み込んで、各レコードに対して実際のクエリを生成する。このとき、各レコードの更新の種類に応じて当該種類のクエリテンプレート（例えば、入力クエリテンプレート、更新クエリテンプレート、削除クエリテンプレートなど）を選択し、選択されたクエリテンプレートにおいてマッピング変数を各レコードデータの当該カラム値にマッピングして実際のクエリ文を生成する。

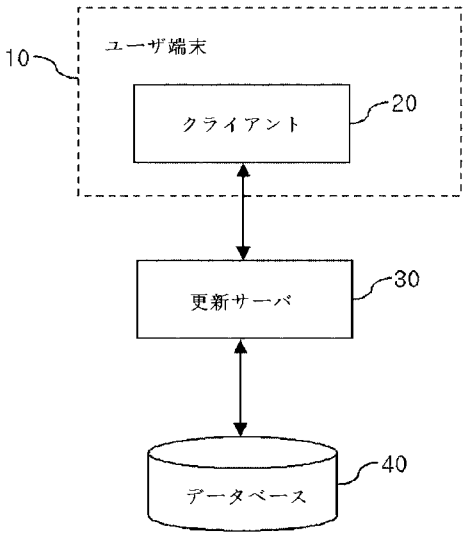
【００８０】
特に、クエリ生成部３５は、更新すべきデータをストリームとして受信し、ストリームとして受信される順番に、各更新すべきレコードの内容に従いクエリ文をリアルタイムにて生成する。
【００８１】
図９に示すように、更新データ又は更新データファイルには、作業の種類（入力Ｃ、更新Ｕ、削除Ｄなど）及び各マッピングされるべきデータ（又は、フィールド値）が連続して記載される。クエリ生成部３５は、各データをレコード別に読み込んで、作業の種類が何であるかを判断して、クエリテンプレートを選択する。また、今後の更新データ（又は、フィールド値）を選択されたクエリテンプレートにおいてマッピング変数に置き換えてマッピングする。このような過程を経て実際のクエリ文が生成される。
【００８２】
次いで、データベース（ＤＢ）更新部３６は、生成された（作成された）クエリ文をデータベース４０に要請して、対象テーブルのデータを更新するように要請する。
【００８３】
データベース（ＤＢ）更新部３６は、前記生成された実際のクエリ文を各レコード別にデータベース４０に転送して更新を要請する。しかしながら、個別的に各レコード別にクエリ文をデータベース４０に要請すると、全体の更新時間が長引き、しかも、コストが嵩む。これは、データベースは一回のトランザクションにより一つの要請を処理するが、トランザクションを行うための作業が多いためである。
【００８４】
このため、好ましくは、データベース（ＤＢ）更新部３６は、多数のレコード別の実際のクエリをバッチでくくって一括処理する。すなわち、クエリ文は一回のみ生成し、転送データの処理は、当該クエリの変数に当該カラム値のみをマッピングしてバッチに追加した後、最後にもう一回データベースに実行を要請するので、速やかに大量のデータを処理することができる。
【００８５】
次いで、本発明の第２の実施形態によるスプレッドシートインタフェースに基づくデータベースの一括更新システムについて図１１を参照して具体的に説明する。本発明の第２の実施形態は、上記の第１の実施形態とその構成が同様であり、但し、更新状態フィールドをワークシート上でカラムに加えて一つの行（ｒｏｗ）に表示する。
【００８６】
図１１に示すように、ワークシート表示部３２は、更新状態を表示する行を更に表示する。図１１において、「ｃｈｅｃｋ２」で表示された行が更新状態を示すレコードである。すなわち、ワークシート上で一つの行を追加して、当該行を更新状態を示すレコードとして用いる。これを第２の更新状態フィールド（又は、第２の更新状態レコード）と称する。上述した第１の実施形態と同様に、更新状態を表示する列の更新状態フィールドを第１の更新状態フィールドと称する。
【００８７】
次いで、ワークシート更新部３３は、ワークシート上で一つのデータフィールド値（又は、セル値）が変更（更新）されると、当該フィールドに対応する第１の更新フィールド及び第２の更新フィールドの両方ともに更新状態を表示する。このとき、第２の更新フィールドにはデータ値（又は、セル値）が変更される場合に表示される。また、ワークシート上で一つのレコードが追加されて挿入される場合（新たなレコードが追加される場合）には、新たな値が入力されるセルに対応する第２の更新フィールドが「Ｃ」などに更新されることが表示される。このため、いかなる値も記載されていないため空きセルの状態で残っているフィールドに対しては、対応する第２の更新フィールドに更新状態が表示されない。また、レコードが削除される場合には、第２の更新フィールドの内容は「Ｕ」や「Ｃ」などの表示に更新されて表示されない。すなわち、第２の更新フィールドの値を変更しない。

【００８８】
図１１の例において、データ値が変更されたものは、キー１０５、１０６、１０８のレコードのうち給料ｓａｌａｒｙ及び電話番号ｐｈｏｎｅ＿ｎｕｍｂｅｒのフィールドである。このため、これらの２つのフィールドに対応する第２の更新フィールドのみ「Ｕ」に更新されて表示される。
【００８９】
次いで、更新定義部３４は、ワークシートの上に第２の更新フィールドのうち「Ｕ」で表示されるフィールド（カラム）及びキーフィールドのみを組み合わせて、データセットの構造又はスキーマの構造を生成する。
【００９０】
図１１の例において、キーフィールドは、「雇用者ＩＤＥＭＰＬＯＹＥＥ＿ＩＤ」であり、変更されたフィールドは給料ｓａｌａｒｙ及び電話番号ｐｈｏｎｅ＿ｎｕｍｂｅｒである。このため、これらのフィールド（又は、カラム）に対してのみデータセットの構造又はスキーマの構造を生成する。前記例に従い作成されたデータセットの構造のＸＭＬドキュメントは、図１２に示す通りである。
【００９１】
図１２に示すように、カラムフィールドは、雇用者ＩＤと、電話番号及び年給に対するフィールドのみをｘｍｌドキュメントに対してのみ示す。
【００９２】
また、更新定義部３４は、更新されたデータ又はレコードを生成するとき、更新されたデータセットの構造のみを表示する。すなわち、更新すべきデータを予め定められたフォーマットで書き込んでファイルとして生成するとき、キーフィールド及び更新されたフィールドに対応するフィールド値（カラム値）のみを表示して、フラットファイルとして構成する。その例が図１３に示されている。
【００９３】
すなわち、図１３に示すように、更新されたフィールドには、キーフィールドであるＥＭＰＬＯＹＥＥ＿ＩＤと、更新されたフィールドであるＰＨＯＮＥ＿ＮＵＭＢＥＲ及びＳＡＬＡＲＹフィールドのデータ値（カラム値）のみが書き込まれる。また、更新されたデータは、雇用者ＩＤＥＭＰＬＯＹＥＥ＿ＩＤが１０４、１０５、１０６、１０８といずれも４つである。このため、４つの更新データ又は更新レコードのみが更新データ（又は、更新データファイル）に書き込まれる。
【００９４】
次いで、クエリ生成部３５は、更新すべきデータのデータセットの構造を用いて、更新されたデータに対するクエリ文を生成する。このとき、クエリ生成部３５は、データセットの構造を用いてクエリテンプレートを予め生成しておき、更新すべきデータのそれぞれに対してクエリテンプレートにマッピングして実際に更新するクエリを生成する。
【００９５】
特に、クエリ生成部３５は、更新フィールドのみを示すデータセットの構造を用いてクエリテンプレートを生成する。図１４は、クエリテンプレートの例を示している。図１４の（ａ）又は（ｂ）に示すように、カラム名は、ＥＭＰＬＯＹＥＥ＿ＩＤ、ＰＨＯＮＥ＿ＮＵＭＢＥＲ、ＳＡＬＡＲＹでのみ表示される。すなわち、キーフィールド及び更新されたフィールドのみにより構成される。
【００９６】
また、クエリ生成部３５は、データをそれぞれレコード別に読み込んで、各レコードに対して実際のクエリを生成する。このとき、各レコードの更新の種類に応じて当該種類のクエリテンプレート（例えば、入力クエリテンプレート、更新クエリテンプレート、削除クエリテンプレートなど）を選択し、選択されたクエリテンプレートにおいてマッピング変数を各レコードデータの当該カラム値にマッピングして実際のクエリ文を生成する。
【００９７】
図１５は、図１３の更新データ（又は、レコード）に対して作成された実際のクエリを

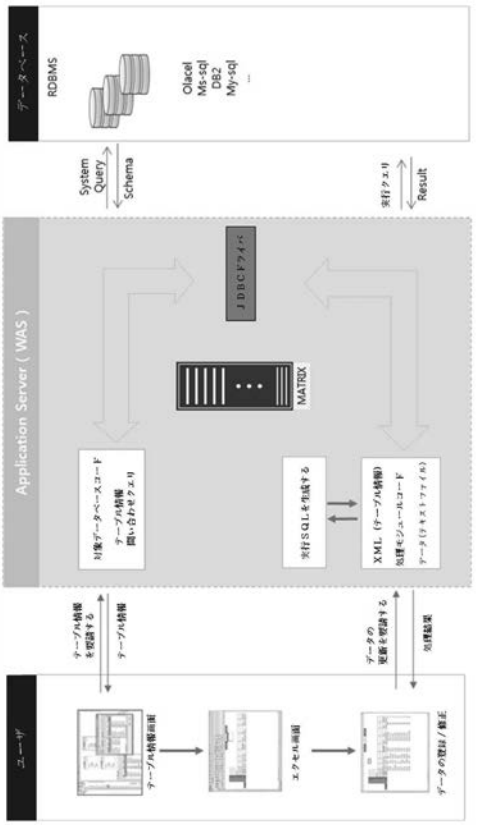
示す。
【００９８】
次いで、本発明の効果についてより具体的に説明する。
【００９９】
まず、本発明は、実行件数に制約がない。
【０１００】
ユーザが登録／修正／削除したデータをアップロードストリームを介してサーバに転送するので、単件データから数百万件の大量データに至るまで自由に取り扱うことができる。データ情報は、定められた行、列の区切り子により区分されたフラットファイとして作成されてサーバにアップロードされ、サーバは、アップロードストリームからデータを読み込んで定められた行、列の区切り子を用いてパーシングした後に、一緒に転送されたｘｍｌ情報から予め生成されたクエリにバインドして実行する。
【０１０１】
また、大量のデータを速やかに処理することができる。
【０１０２】
更新サーバにおいてデータを処理する基本方式は、ｊｄｂｃドライバを介してデータベースに接続した後、クライアントから送られてきたｘｍｌ情報をパーシングして入力、修正、削除に相当するクエリ文を生成し、一緒にクライアントから送られてきたアップロードストリームからデータを読み込みながら予め生成されたクエリに変数をバインドした後に実行することである。
【０１０３】
このとき、接続客体のうちＰｒｅｐａｒｅｄＳｔａｔｅｍｅｎｔを用いて大きさ（フェッチサイズ）を指定し、バインドされたデータをＰｒｅｐａｒｅｄＳｔａｔｅｍｅｎｔのバッチ手段ａｄｄＢａｔｃｈｍｅｔｈｏｄを用いて追加し、全てのデータが追加されたときに一括して実行してデータを処理する。毎件ごとに実行する方式ではなく、一括して実行する方式であるので、データベースの負荷も減り、しかも、速度も非常に速い。
【０１０４】
また、データベース（ＤＢ）機能を使用機能として提供する。
【０１０５】
ユーザが画面において処理したデータだけではなく、データベース機能、ユーザ生成関数などを処理するようにオプションを提供して、サーバにおいてクエリを生成するときに対象カラム処理をデータバインドの代わりに関数処理する機能を提供する。
【０１０６】
また、オラクルシーケンス、ｓｙｓｄａｔｅなど時間関数、ｒｏｗｎｕｍなどのＤＢ関数などをクエリに含めて用いる。
【０１０７】
以上、本発明者により行われた発明について実施形態により具体的に説明したが、本発明はこれらの実施形態に何ら限定されるものではなく、その要旨を逸脱しない範囲内において種々に変更可能であるということはいうまでもない。
参考までに、本発明を支援した韓国の国家研究開発事業を表示すると、下記の通りである。
［課題の固有番号］　１００５３２０４
［部処名］　産業通商資源部
［研究管理専門機関］　韓国産業技術評価管理院
［研究事業名］　産業技術革新事業（グローバル融合ＡＴＣ事業）
［研究課題名］　５億件のデータを基準として３秒以内に「データ処理及びユーザインタフェースの実現」が可能なペタバイト級非専門家用ビジネスインテリジェンスサービスプラットフォーム技術開発
［主観機関］　株式会社ＢＩマトリックス
［研究期間］２０１５年６月１日～２０２０年５月３１日

【符号の説明】
【０１０８】
１０：ユーザ端末
２０：クライアント
３０：更新サーバ
３１：データ要請部
３２：ワークシート表示部
３３：ワークシート更新部
３４：更新定義部
３５：クエリ生成部
３６：データベース（ＤＢ）更新部
３７：保存部
４０：データベース

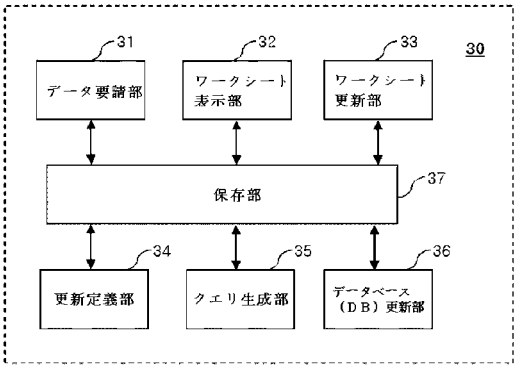
【図１】



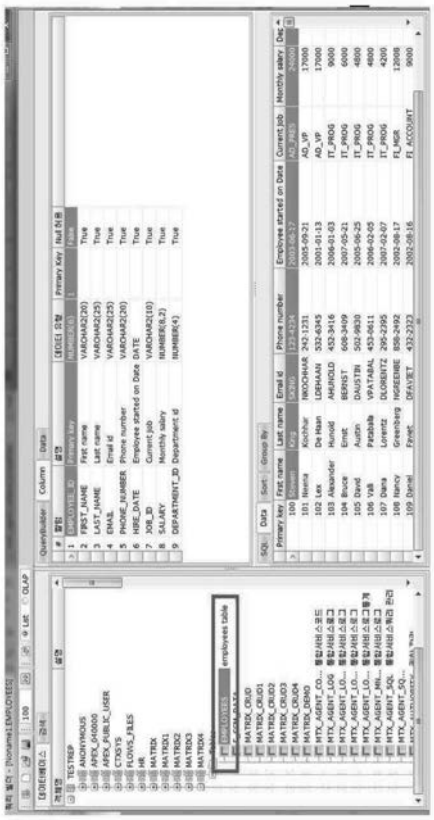
【図２】



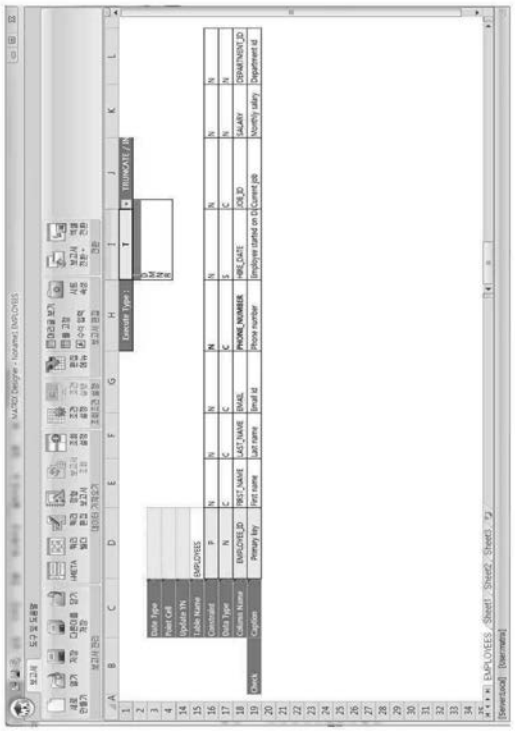
【図３】


31
32
33
30
データ要請部
ワークシート表示部
ワークシート更新部
保存部
37
34
35
36
更新定義部
クエリ生成部
データベース（ＤＢ）更新部

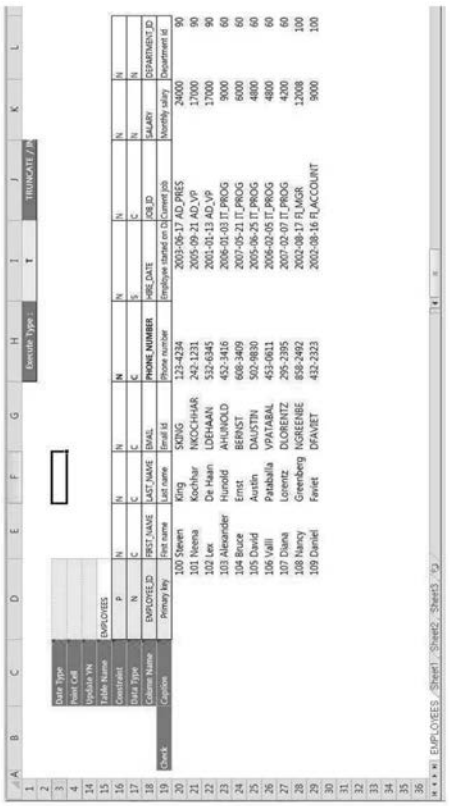
【図４】



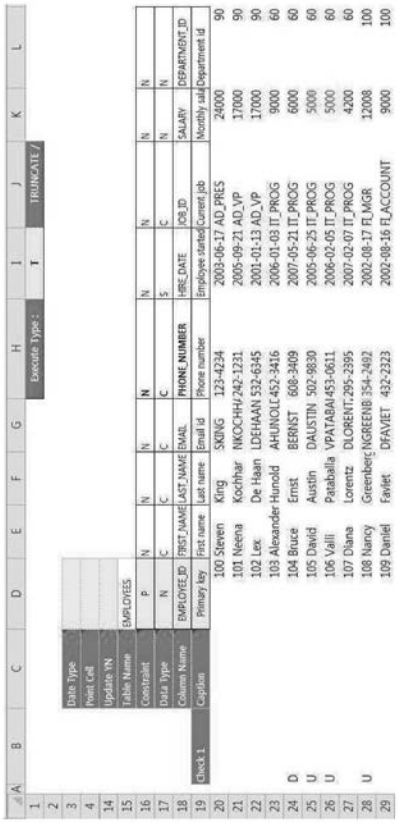
【図５】



【図６】



【図７】



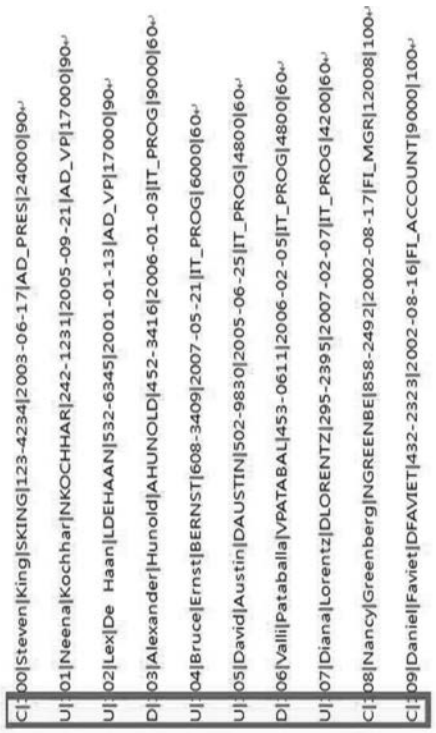
	A	B	C	D	E	F	G	H	I	J	K	L
1												
2								Execute Type :	T	TRUNCATE /		
3			Date Type									
4			Point Cell									
14			Update YN									
15			Table Name	EMPLOYEES								
16			Constraint	P	N	N	N	N	N	N	N	N
17			Data Type	N	C	C	C	C	S	C	N	N
18			Column Name	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID
19		Check 1	Caption	Primary key	First name	Last name	Email id	Phone number	Employee started	Current job	Monthly sala	Department id
20				100	Steven	King	SKING	123-4234	2003-06-17	AD_PRES	24000	90
21				101	Neena	Kochhar	NKOCHHA	242-1231	2005-09-21	AD_VP	17000	90
22				102	Lex	De Haan	LDEHAAN	532-6345	2001-01-13	AD_VP	17000	90
23				103	Alexander	Hunold	AHUNOLD	452-3416	2006-01-03	IT_PROG	9000	60
24	D			104	Bruce	Ernst	BERNST	608-3409	2007-05-21	IT_PROG	6000	60
25	U			105	David	Austin	DAUSTIN	502-9830	2005-06-25	IT_PROG	5000	60
26	U			106	Valli	Pataballa	VPATABAL	453-0611	2006-02-05	IT_PROG	5000	60
27				107	Diana	Lorentz	DLORENTZ	295-2395	2007-02-07	IT_PROG	4200	60
28	U			108	Nancy	Greenberg	NGREENBE	354-2492	2002-08-17	FI_MGR	12008	100
29				109	Daniel	Faviet	DFAVIET	432-2323	2002-08-16	FI_ACCOUNT	9000	100

【図８】



```
<?xml version="1.0" encoding="utf-8"?>
<TransPacket>
<AuthInfo UserCode="matrix" />
<ExecuteInfo ConnectionCode="DBDE1037DF740A5119390ACA4FC5D1FF79" ExecuteType="M"/>
<DataSet Name="" Caption="">
<DataTables>
<DataTable Name="EMPLOYEES" Caption="EMPLOYEES" TargetTable="EMPLOYEES">
<Columns>
<Column Name="EMPLOYEE_ID" Caption="EMPLOYEE_ID" Constraint="P" DataType="C" />
<Column Name="FIRST_NAME" Caption="FIRST_NAME" Constraint="N" DataType="C" UpdateFlag="Y" />
<Column Name="LAST_NAME" Caption="LAST_NAME" Constraint="N" DataType="C" UpdateFlag="Y" />
<Column Name="EMAIL" Caption="EMAIL" Constraint="N" DataType="C" UpdateFlag="Y" />
<Column Name="PHONE_NUMBER" Caption="PHONE_NUMBER" Constraint="N" DataType="C" UpdateFlag="Y" />
<Column Name="HIRE_DATE" Caption="HIRE_DATE" Constraint="N" DataType="C" UpdateFlag="Y" />
<Column Name="JOB_ID" Caption="JOB_ID" Constraint="N" DataType="C" UpdateFlag="Y" />
<Column Name="SALARY" Caption="SALARY" Constraint="N" DataType="C" UpdateFlag="Y" />
<Column Name="DEPARTMENT_ID" Caption="DEPARTMENT_ID" Constraint="N" DataType="C" UpdateFlag="Y" />
</Columns>
<Rows RowIndexCol="0" RowStateCol="1" TextType="FILE"></Rows>
</DataTable>
</DataTables>
</DataSet>
</TransPacket>
```

【図９】



```
C|00|Steven|King|SKING|123-4234|2003-06-17|AD_PRES|24000|90
U|01|Neena|Kochhar|NKOCHHAR|242-1231|2005-09-21|AD_VP|17000|90
U|02|Lex|De  Haan|LDEHAAN|532-6345|2001-01-13|AD_VP|17000|90
D|03|Alexander|Hunold|AHUNOLD|452-3416|2006-01-03|IT_PROG|9000|60
U|04|Bruce|Ernst|BERNST|608-3409|2007-05-21|IT_PROG|6000|60
U|05|David|Austin|DAUSTIN|502-9830|2005-06-25|IT_PROG|4800|60
D|06|Valli|Pataballa|VPATABAL|453-0611|2006-02-05|IT_PROG|4800|60
U|07|Diana|Lorentz|DLORENTZ|295-2395|2007-02-07|IT_PROG|4200|60
C|08|Nancy|Greenberg|NGREENBE|858-2492|2002-08-17|FI_MGR|12008|100
C|09|Daniel|Faviet|DFAVIET|432-2323|2002-08-16|FI_ACCOUNT|9000|100
```

【図１０】

```
INSERT INTO EMPLOYEES ( EMPLOYEE_ID,  FIRST_NAME,
LAST_NAME, EMAIL, PHONE_NUMBER, HIRE_DATE, JOB_ID, SALARY,
DEPARTMENT_ID )
VALUES (@D1, @D2, @D3, @D4, @D5, @D6, @D7, @D8, @D9 )
```

(a)

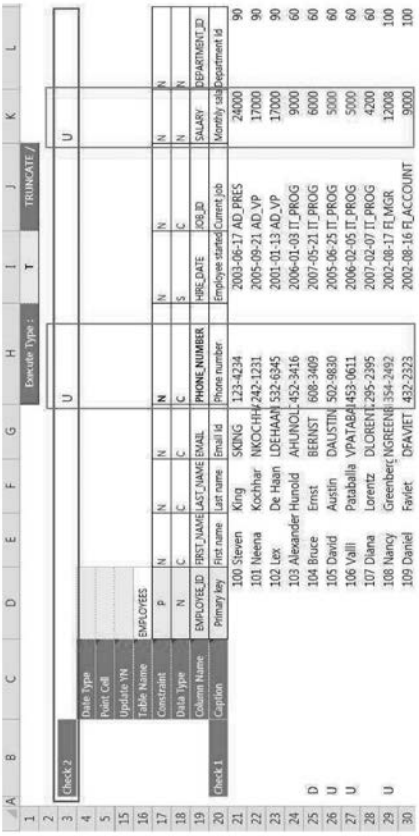
```
UPDATE EMPLOYEES SET FIRST_NAME = @D2 , LAST_NAME =
@D3 , EMAIL = @D4 , PHONE_NUMBER = @D5 , HIRE_DATE = @D6,
JOB_ID = @D7 , SALARY = @D8 , DEPARTMENT_ID = @D9
WHERE EMPLOYEE_ID = @D1
```

(b)

```
DELETE FROM  EMPLOYEES
WHERE EMPLOYEE_ID = @D1
```

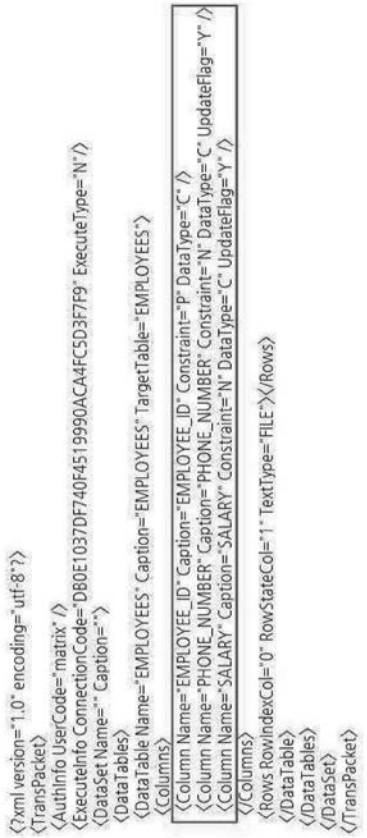
(c)

【図１１】



	A	B	C	D	E	F	G	H	I	J	K	L
1								Execute Type :	T	TRUNCATE /		
2											U	
3		Check 2						U				
4			Date Type									
5			Paint Cell									
15			Update YN									
16			Table Name	EMPLOYEES								
17			Constraint	P	N	N	N	N	N	N	N	N
18			Data Type	N	C	C	C	C	S	C	N	N
19			Column Name	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID
20		Check 1	Caption	Primary key	First name	Last name	Email Id	Phone number	Employee started	Current job	Monthly sala	Department Id
21					100	Steven	King	SKING	123-4234	2003-06-17	AD_PRES	24000	90
22					101	Neena	Kochhar	NKOCHHA	242-1231	2005-09-21	AD_VP	17000	90
23					102	Lex	De Haan	LDEHAAN	532-6345	2001-01-13	AD_VP	17000	90
24					103	Alexander	Hunold	AHUNOLD	452-3416	2006-01-03	IT_PROG	9000	60
25		D			104	Bruce	Ernst	BERNST	608-3409	2007-05-21	IT_PROG	6000	60
26		U			105	David	Austin	DAUSTIN	502-9830	2005-06-25	IT_PROG	5000	60
27		U			106	Valli	Pataballa	VPATABAL	453-0611	2006-02-05	IT_PROG	5000	60
28					107	Diana	Lorentz	DLORENTZ	295-2395	2007-02-07	IT_PROG	4200	60
29		U			108	Nancy	Greenberg	NGREENB	354-2492	2002-08-17	FI_MGR	12008	100
30					109	Daniel	Faviet	DFAVIET	432-2323	2002-08-16	FI_ACCOUNT	9000	100

【図１２】



```
<?xml version="1.0" encoding="utf-8"?>
<TransPacket>
<AuthInfo UserCode="matrix" />
<ExecuteInfo ConnectionCode="DB0E1037DF740F4519990ACA4FC5D3F7F9" ExecuteType="N" />
<DataSet Name="" Caption="">
<DataTables>
<DataTable Name="EMPLOYEES" Caption="EMPLOYEES" TargetTable="EMPLOYEES">
<Columns>
<Column Name="EMPLOYEE_ID" Caption="EMPLOYEE_ID" Constraint="P" DataType="C" />
<Column Name="PHONE_NUMBER" Caption="PHONE_NUMBER" Constraint="N" DataType="C" UpdateFlag="Y" />
<Column Name="SALARY" Caption="SALARY" Constraint="N" DataType="C" UpdateFlag="Y" />
</Columns>
<Rows RowIndexCol="0" RowStateCol="1" TextType="FILE">X</Rows>
</DataTable>
</DataTables>
</DataSet>
</TransPacket>
```

【図１３】

```
D|104|608-3409|6000
U|105|502-9830|5000
U|106|453-0611|5000
U|108|354-2492|12008
```

【図１４】

```
INSERT INTO EMPLOYEES ( EMPLOYEE_ID, PHONE_NUMBER,
SALARY)
VALUES (@D1, @D2, @D3)
```

(a)

```
UPDATE EMPLOYEES SET PHONE_NUMBER = @D2 , SALARY =
@D3
WHERE EMPLOYEE_ID = @D1
```

(b)

```
DELETE FROM EMPLOYEES
WHERE EMPLOYEE_ID = @D1
```

(c)

【図１５】

```
DELETE FROM EMPLOYEES
WHERE EMPLOYEE_ID = 104
```

```
UPDATE EMPLOYEES SET PHONE_NUMBER = 502-9830, SALARY = 5000
WHERE EMPLOYEE_ID = 105
```

```
UPDATE EMPLOYEES SET PHONE_NUMBER = 453-0611, SALARY = 5000
WHERE EMPLOYEE_ID = 106
```

```
UPDATE EMPLOYEES SET PHONE_NUMBER = 354-2492, SALARY = 12008
WHERE EMPLOYEE_ID = 108
```

フロントページの続き

(72)発明者　カン　ジュヒ
大韓民国　ソウル，ウンピョング，チングァン　２ロ，６０，３２７－１００２