

(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2014-106557
(P2014-106557A)

(43) 公開日 平成26年6月9日(2014.6.9)

| | | |
|-----------------------------|-----------------|-------------|
| (51) Int.Cl. | F I | テーマコード (参考) |
| G06F 12/00 (2006.01) | G06F 12/00 513Z | 5B376 |
| G06F 9/44 (2006.01) | G06F 9/06 620K | |

審査請求 未請求 請求項の数 7 O L (全 22 頁)

| | | | |
|-----------|------------------------------|----------|-------------------------------------------------|
| (21) 出願番号 | 特願2012-256729 (P2012-256729) | (71) 出願人 | 390002761 |
| (22) 出願日 | 平成24年11月22日 (2012.11.22) | | キヤノンマーケティングジャパン株式会社 東京都港区港南2丁目16番6号 |
| | | (71) 出願人 | 312000206 |
| | | | キヤノンMJアイティグループホールディングス株式会社 東京都品川区東品川2丁目4番11号 |
| | | (71) 出願人 | 592135203 |
| | | | キヤノンITソリューションズ株式会社 東京都品川区東品川2丁目4番11号 |
| | | (74) 代理人 | 100189751 |
| | | | 弁理士 木村 友輔 |
| | | (74) 代理人 | 100188938 |
| | | | 弁理士 榛葉 加奈子 |

最終頁に続く

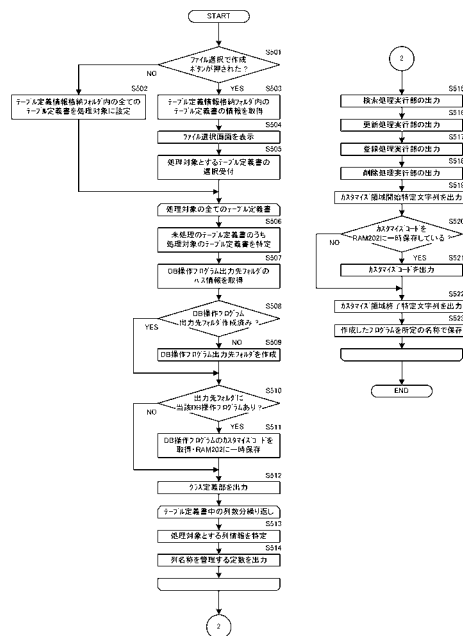
(54) 【発明の名称】 情報処理装置、情報処理方法、及びコンピュータプログラム

(57) 【要約】

【課題】 データベースに登録されているデータを操作するためのプログラムを自動生成する際に、既にプログラムが自動作成されており、そのプログラムに修正を加えられている場合に、その修正を反映させた形で、新たにプログラムを自動作成する。

【解決手段】 データベース操作のためのプログラム作成指示を受け付けると、当該データベース操作のためのプログラムを既に作成しているかを判定する。既に作成していると判定した場合には、そのプログラム中の修正コードを取得し、新たに作成するプログラムに追加する。

【選択図】 図5



【特許請求の範囲】**【請求項 1】**

データベースのデータ操作を行うためのプログラムを生成する情報処理装置であって、前記データベースに登録されているデータテーブルの構成を定義したテーブル定義書の指定を受け付ける受付手段と、

前記受付手段で指定を受け付けたテーブル定義書を用いて前記データテーブルのデータ操作を行うためのプログラムを作成する第 1 の作成手段と、

前記受付手段で指定を受け付けたデータ定義書を用いて、前記第 1 の作成手段により、既に前記データテーブルのデータ操作を行うためのプログラムが作成されているかを判定する判定手段と、

前記判定手段で前記プログラムが既に作成されていると判定される場合に、当該既に作成されているプログラムから、修正されたコードを取得する第 1 の取得手段とを備え、

前記第 1 の作成手段は、前記第 1 の取得手段で取得した修正されたコードを含めた形で、前記データテーブルのデータ操作を行うためのプログラムを作成すること

を特徴とする情報処理装置。

【請求項 2】

前記データベースにアクセスするアクセス手段と、

前記アクセス手段によりアクセスしたデータベースに登録されているデータテーブルから、データテーブルを特定するための情報および、当該データテーブルに含まれる列情報を含むデータテーブル情報を取得する第 2 の取得手段と、

前記第 2 の取得手段で取得したデータテーブル情報を用いて、前記データテーブルの構成を定義したテーブル定義書を作成する第 2 の作成手段と

を備えることを特徴とする請求項 1 に記載の情報処理装置。

【請求項 3】

前記受付手段で指定を受け付けたテーブル定義書を用いて、当該テーブル定義書で定義されるデータテーブルをデータベースに登録するために用いられる DDL を作成する第 3 の作成手段

を更に備えることを特徴とする請求項 1 または 2 に記載の情報処理装置。

【請求項 4】

前記第 1 の作成手段は、前記プログラムに、ソースコードのカスタマイズ領域を特定するための、カスタマイズ領域開始特定文字列及びカスタマイズ領域終了特定文字列を含むプログラムを生成すること

を特徴とする請求項 1 乃至 3 のいずれか 1 項に記載の情報処理装置。

【請求項 5】

前記第 1 の取得手段は、前記カスタマイズ領域開始特定文字列とカスタマイズ領域終了特定文字列とに挟まれたコードを、修正されたコードとして、前記プログラムから取得すること

を特徴とする請求項 4 に記載の情報処理装置。

【請求項 6】

データベースのデータ操作を行うためのプログラムを生成する情報処理装置によって行われる情報処理方法であって、

前記データベースに登録されているデータテーブルの構成を定義したテーブル定義書の指定を受け付ける受付工程と、

前記受付工程で指定を受け付けたテーブル定義書を用いて前記データテーブルのデータ操作を行うためのプログラムを作成する第 1 の作成工程と、

前記受付工程で指定を受け付けたデータ定義書を用いて、前記第 1 の作成工程により、既に前記データテーブルのデータ操作を行うためのプログラムが作成されているかを判定する判定工程と、

前記判定工程で前記プログラムが既に作成されていると判定される場合に、当該既に作成されているプログラムから、修正されたコードを取得する第 1 の取得工程とを備え、

10

20

30

40

50

前記第 1 の作成工程は、前記第 1 の取得工程で取得した修正されたコードを含めた形で、前記データテーブルのデータ操作を行うためのプログラムを作成することを特徴とする情報処理方法。

【請求項 7】

データベースのデータ操作を行うためのプログラムを生成する情報処理装置を、前記データベースに登録されているデータテーブルの構成を定義したテーブル定義書の指定を受け付ける受付手段と、

前記受付手段で指定を受け付けたテーブル定義書を用いて前記データテーブルのデータ操作を行うためのプログラムを作成する第 1 の作成手段と、

前記受付手段で指定を受け付けたデータ定義書を用いて、前記第 1 の作成手段により、既に前記データテーブルのデータ操作を行うためのプログラムが作成されているかを判定する判定手段と、

前記判定手段で前記プログラムが既に作成されていると判定される場合に、当該既に作成されているプログラムから、修正されたコードを取得する第 1 の取得手段として機能させ、

前記第 1 の作成手段は、前記第 1 の取得手段で取得した修正されたコードを含めた形で、前記データテーブルのデータ操作を行うためのプログラムを作成することを特徴とするコンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、データベース操作を行うためのプログラムの生成に用いるソースコードを自動生成する技術に関する。

【背景技術】

【0002】

データベースシステムを構築する際には、データベースへの新規データの登録、データベースに登録されているデータの検索、データベースに登録されているデータの更新や削除等を行うためのプログラムを開発する必要がある。

【0003】

このようなデータベース操作のためのプログラムを作成するためには、SQL等のデータベース操作言語に習熟している必要があるが、一般の人々が習得するには困難が伴う。また、このプログラムは、データベースの構成に変更が入るたびに作成する必要があり、プログラムの開発者はその度にプログラムの更新を行わなくてはならなかった。

【0004】

そこで、特許文献 1 には、特許文献 2 には、スキーマを記述するファイルなどを利用して、アプリケーションプログラム自動生成する発明が開示されている。

【先行技術文献】

【特許文献】

【0005】

【特許文献 1】特開 2006 - 004145 号公報

【発明の概要】

【発明が解決しようとする課題】

【0006】

上記の先行技術文献に記載の発明により、プログラムの自動生成を行うことは可能となる。しかし、全てのデータベースシステムに必要な機能を具備するプログラムを自動生成により作成することは困難である。そのため、このように自動生成されたプログラム（ソースコード）に、プログラム開発者がデータベースシステム毎に個別に必要な機能を具備させるためのソースコードを追加することが通常である。

【0007】

しかし、このようなプログラム（ソースコード）を自動生成する場合には、プログラム

10

20

30

40

50

ファイル全体を新規ファイルとして作成し直すために、ソースコードの自動生成を再度行う場合に、過去に自動生成されたプログラムにプログラム開発者が追加したソースコードが、新たに作成されるプログラムに反映されず、プログラム開発者が、再度そのソースコードを追加しなくてはならなかった。また、場合によっては、追加したソースコードが消失してしまうという恐れもあった。

【0008】

そこで、本発明は、自動生成されたプログラムに変更を加えた後に、再度プログラムの自動生成処理を行った場合に、過去に自動生成プログラムに行った変更内容を反映した形でプログラムを自動生成する技術を提供することを目的とする。

【課題を解決するための手段】

【0009】

上記した目的を達成するために、本発明の情報処理装置は、データベースのデータ操作を行うためのプログラムを生成する情報処理装置であって、前記データベースに登録されているデータテーブルの構成を定義したテーブル定義書の指定を受け付ける受付手段と、前記受付手段で指定を受け付けたテーブル定義書を用いて前記データテーブルのデータ操作を行うためのプログラムを作成する第1の作成手段と、前記受付手段で指定を受け付けたデータ定義書を用いて、前記第1の作成手段により、既に前記データテーブルのデータ操作を行うためのプログラムが作成されているかを判定する判定手段と、前記判定手段で前記プログラムが既に作成されていると判定される場合に、当該既に作成されているプログラムから、修正されたコードを取得する第1の取得手段とを備え、前記第1の作成手段は、前記第1の取得手段で取得した修正されたコードを含めた形で、前記データテーブルのデータ操作を行うためのプログラムを作成することを特徴とする。

【0010】

上記した目的を達成するために、本発明の情報処理方法は、データベースのデータ操作を行うためのプログラムを生成する情報処理装置によって行われる情報処理方法であって、前記データベースに登録されているデータテーブルの構成を定義したテーブル定義書の指定を受け付ける受付工程と、前記受付工程で指定を受け付けたテーブル定義書を用いて前記データテーブルのデータ操作を行うためのプログラムを作成する第1の作成工程と、前記受付工程で指定を受け付けたデータ定義書を用いて、前記第1の作成工程により、既に前記データテーブルのデータ操作を行うためのプログラムが作成されているかを判定する判定工程と、前記判定工程で前記プログラムが既に作成されていると判定される場合に、当該既に作成されているプログラムから、修正されたコードを取得する第1の取得工程とを備え、前記第1の作成工程は、前記第1の取得工程で取得した修正されたコードを含めた形で、前記データテーブルのデータ操作を行うためのプログラムを作成することを特徴とする。

【0011】

上記した目的を達成するために、本発明のコンピュータプログラムは、データベースのデータ操作を行うためのプログラムを生成する情報処理装置を、前記データベースに登録されているデータテーブルの構成を定義したテーブル定義書の指定を受け付ける受付手段と、前記受付手段で指定を受け付けたテーブル定義書を用いて前記データテーブルのデータ操作を行うためのプログラムを作成する第1の作成手段と、前記受付手段で指定を受け付けたデータ定義書を用いて、前記第1の作成手段により、既に前記データテーブルのデータ操作を行うためのプログラムが作成されているかを判定する判定手段と、前記判定手段で前記プログラムが既に作成されていると判定される場合に、当該既に作成されているプログラムから、修正されたコードを取得する第1の取得手段として機能させ、前記第1の作成手段は、前記第1の取得手段で取得した修正されたコードを含めた形で、前記データテーブルのデータ操作を行うためのプログラムを作成することを特徴とする。

【発明の効果】

【0012】

本発明によれば、過去に自動生成したプログラムの自動生成処理を再度行う場合に、過

10

20

30

40

50

去に自動生成したプログラムに加えられた変更を反映させた形で新たなプログラムの自動生成を行うことが可能となる。

【図面の簡単な説明】

【0013】

【図1】本発明の情報処理装置の機能構成の一例を示す図である。

【図2】本発明の情報処理装置のハードウェア構成の一例を示す図である。

【図3】情報処理装置100のCPU201によって行われる各種ファイル作成処理の概要を示すフローチャートである。

【図4】図3のステップS308のテーブル定義書作成処理の詳細を示すフローチャートである。

【図5】図3のステップS310のDB操作プログラム作成処理の詳細を示すフローチャートである。

【図6】図3のステップS312のDDL作成処理の詳細を示すフローチャートである。

【図7】操作指示画面の構成の一例を示す図である。

【図8】システム定義情報の構成の一例を示す図である。

【図9】テーブル定義書の構成の一例を示す図である。

【図10】DB操作プログラムのソースコード中のカスタマイズ領域の一例を示す図である。

【図11】プログラム出力部104により出力されるDB操作プログラムのクラス定義部のソースコードの一例を示す図である。

【図12】プログラム出力部104により出力されるDB操作プログラムの定数定義部のソースコードの一例を示す図である。

【図13】プログラム出力部104により出力されるDB操作プログラムの処理実行部のソースコードの一例を示す図である。

【図14】DDL出力部105により出力されるDDLの構成の一例を示す図である。

【図15】DDL出力部105により出力されるDDLの構成の一例を示す図である。

【図16】DDL出力部105により出力されるDDLの構成の一例を示す図である。

【発明を実施するための形態】

【0014】

以下、図面を参照して本発明の実施の形態の一例について説明する。

【0015】

図1は、本発明の情報処理システムの機能構成の一例を示す図である。図1に示す通り、本発明の情報処理装置100は、テーブル情報入力部101、テーブル定義書出力部102、テーブル定義書入力部103、プログラム出力部104、DDL出力部105等を備えて構成されている。

【0016】

テーブル情報入力部101は、データベース111にアクセスし、データベース111に登録されているデータテーブル情報を入力するための機能部である。テーブル定義書出力部102は、テーブル情報入力部102により入力されたデータベース111のテーブル情報を用いて、図9に示すテーブル定義書を出力する機能部である。

【0017】

テーブル定義書入力部103は、所定のフォルダに記憶されているテーブル定義書に登録されている情報を入力する機能部である。

【0018】

プログラム出力部104は、テーブル定義書入力部103により入力されたテーブル定義情報を用いて、データベース(DB)操作プログラム113を自動生成する機能部である。尚、プログラム出力部104は、同一のテーブル定義書から過去に作成したプログラムがある場合に、そのプログラムへのユーザによる変更を反映した形で、新たにDB操作プログラムを作成することになる。

【0019】

10

20

30

40

50

例えば、自動生成されたプログラム（ソースコード）中のカスタマイズ領域に追加されたソースコードを、新たに作成されるDB操作プログラムのソースコードにも含める形で、DB操作プログラムを生成する。

【0020】

DDL作成部105は、テーブル定義書入力部103により入力されたテーブル定義情報を用いて、データテーブル111のデータテーブルの作成を行うためのDDL（Data Definition Language）を作成する機能部である。

【0021】

本発明の情報処理装置100は以上のような構成を備えるために、初めにデータ定義書を作成しておけば、当該データ定義書で定義されたデータテーブル、そのデータテーブルに登録されるデータを操作するためのDB操作プログラム113、及び当該データテーブルと同様の構成のデータテーブルの作成に用いられるDDL114の作成が可能となる。

10

【0022】

また、データテーブルを作成すれば、当該データテーブルのテーブル定義書112、当該データテーブルに登録されているデータを操作するためのDB操作プログラム113、及び当該データテーブルと同様の構成のデータテーブルの作成に用いられるDDL114の作成が可能となる。

【0023】

以上が、本発明の情報処理装置の機能構成の一例の説明である。

【0024】

次に、図2を参照して、本発明の情報処理装置（図1の情報処理装置100）のハードウェア構成の一例について説明する。

20

【0025】

図2において、201はCPUで、システムバス204に接続される各デバイスやコントローラを統括的に制御する。また、ROM203あるいは外部メモリ211には、CPU201の制御プログラムであるBIOS（Basic Input / Output System）やオペレーティングシステムプログラム（以下、OS）や、情報処理装置100が実行する機能を実現するために必要な後述する各種プログラム等が記憶されている。

【0026】

202はRAMで、CPU201の主メモリ、ワークエリア等として機能する。CPU201は、処理の実行に際して必要なプログラム等をROM203あるいは外部メモリ211からRAM202にロードして、該ロードしたプログラムを実行することで後述する各種動作を実現するものである。

30

【0027】

また、205は入力コントローラで、キーボードやポインティングデバイス等の入力装置209からの入力を制御する。206はビデオコントローラで、ディスプレイ装置210等の表示器への表示を制御する。これらは必要に応じて操作者が使用するものである。

【0028】

207はメモリコントローラで、ブートプログラム、各種のアプリケーション、フォントデータ、ユーザファイル、編集ファイル、各種データ等を記憶するハードディスク（HD）や、フレキシブルディスク（FD）、或いはPCMCIAカードスロットにアダプタを介して接続されるコンパクトフラッシュ（登録商標）メモリ等の外部メモリ211へのアクセスを制御する。

40

【0029】

208は通信I/F（インタフェース）コントローラで、ネットワーク（例えば、図1に示したネットワーク104）を介して外部機器と接続・通信するものであり、ネットワークでの通信制御処理を実行する。例えば、TCP/IPを用いた通信等が可能である。

【0030】

なお、CPU201は、例えばRAM202内の表示情報用領域へアウトラインフォントの展開（ラスターライズ）処理を実行することにより、ディスプレイ装置210上での表

50

示を可能としている。また、CPU 201は、ディスプレイ装置 210上の不図示のマウスカーソル等でのユーザ指示を可能とする。

【0031】

本発明を実現するための後述するフローチャートに示す各ステップの処理は、コンピュータで読み取り実行可能なプログラムにより実行され、そのプログラムは外部メモリ 211に記録されている。そして、必要に応じてRAM 202にロードされることによりCPU 201によって実行されるものである。これにより、CPU 201は、図1に示す各機能部として動作し、フローチャートに示す各ステップの処理を行うことになる。さらに、上記プログラムの実行時に用いられる定義ファイル及び各種情報テーブル等も、外部メモリ 211に格納されており、これらについての詳細な説明も後述する。

10

【0032】

次に、図3を参照して、情報処理装置 100のCPU 201によって行われる各種ファイル作成処理の概要について説明する。

【0033】

情報処理装置 100のCPU 201は、データベース関連ファイル作成アプリケーションの起動指示を受け付けることで入力されるデータベース関連ファイルのファイル作成指示を受け付けると、所定のディレクトリに保存されている設定ファイルの読み込み処理を行う(ステップS301)。このステップでは、図8に示す設定ファイルの読み込み処理を行う。

【0034】

ここで、図8を参照して、図3のステップS301で情報処理装置 100のCPU 201が読み込む設定ファイル 800の構成の一例について説明する。

20

【0035】

図8に示すように、設定ファイル 800には、データベースサーバへの接続を行うための情報であるデータベース接続情報として、データベースサーバのIPアドレス 801、データベースサーバへの接続時の認証に使用するユーザID 802、パスワード 803が記憶されている。

【0036】

また後述する処理で、読み込み処理や保存処理を行うテーブル定義書、DB(データベース)操作プログラム、DDLの格納先であるフォルダの情報がそれぞれ記憶されている(データ定義書格納先フォルダ 804、DB操作プログラム格納先フォルダ 805、DDL格納先フォルダ 806)。以上が、設定ファイル 800の構成の一例の説明である。

30

【0037】

図3の説明に戻る。情報処理装置 100のCPU 201は、ステップS301で設定ファイル 800を読み込むことで、データベース接続情報を取得した後に、取得したデータベース接続情報を用いてデータベースサーバにアクセスする。その後、データベースをオープンし(ステップS302)、データベースに登録されているデータテーブルの一覧を取得する(ステップS303)。その後、データベースをクローズする(ステップS304)。

【0038】

データベースサーバよりデータテーブルのデータを取得した後に、情報処理装置 100のCPU 201は処理をステップS305に進め、図7に示す操作画面 700をディスプレイ装置 210に表示する。

40

【0039】

ここで、図7を参照して、図3のステップS305でディスプレイ装置 210に表示される操作画面 700の構成の一例について説明する。

【0040】

図中701は、データテーブル一覧表示部であって、データベースサーバより取得したデータテーブルの一覧を選択可能に表示する表示部である。702は、データテーブル一覧表示部 701で選択状態のデータテーブルのテーブル定義書の作成指示を、情報処理装

50

置 100 の CPU 201 に入力するために用いられる作成ボタンである。

【0041】

703、704 は、テーブル定義書格納先フォルダに格納されているテーブル定義書から、データベースの操作プログラム（DB 操作プログラム）を作成する指示を、情報処理装置 100 の CPU 201 に入力するために用いられるボタンである。

【0042】

尚、作成ボタン 703 に対する押下指示を受け付けた場合には、情報処理装置 100 の CPU 201 は、不図示のファイル選択画面をディスプレイ装置 210 に表示し、そのファイル選択画面を介して、DB 操作プログラムの作成に用いるテーブル定義書の選択を受け付ける。そして、選択されたテーブル定義書を用いて DB 操作プログラムを作成することになる。

10

【0043】

一方、規定フォルダ内の全ての定義ファイルボタン 704 に対する押下指示を受け付けた場合には、情報処理装置 100 の CPU 201 は、テーブル定義書格納先フォルダに格納されている全てのテーブル定義書を用いて、DB 操作プログラムを作成する。

【0044】

705、706 は、テーブル定義書格納先フォルダに格納されているテーブル定義書から、DDL（Data Definition Language）を作成する指示を、情報処理装置 100 の CPU 201 に入力するために用いられるボタンである。

【0045】

尚、作成ボタン 705 に対する押下指示を受け付けた場合には、情報処理装置 100 の CPU 201 は、不図示のファイル選択画面をディスプレイ装置 210 に表示し、そのファイル選択画面を介して、DDL の作成に用いるテーブル定義書の選択を受け付ける。そして、選択されたテーブル定義書を用いて DB 操作プログラムを作成することになる。

20

【0046】

一方、規定フォルダ内の全ての定義ファイルボタン 706 に対する押下指示を受け付けた場合には、情報処理装置 100 の CPU 201 は、テーブル定義書格納先フォルダに格納されている全てのテーブル定義書を用いて、DDL を作成する。

【0047】

707 は終了ボタンであって、情報処理装置 100 の CPU 201 に本操作画面の表示終了指示を入力するために用いられるボタンである。以上が、図 7 の操作画面 700 の構成の一例の説明である。

30

【0048】

図 3 の説明に戻る。ステップ S 305 で、図 7 に示す操作画面 700 をディスプレイ装置 210 に表示した後に、操作画面 700 を介して、ユーザからの各種の操作指示を受け付ける（ステップ S 306）。

【0049】

情報処理装置 100 の CPU 201 は、ステップ S 306 で、作成ボタン 702 に対する押下指示を受け付けることで入力されるテーブル定義書作成指示を受け付けたと判定した場合には（ステップ S 307 で YES）、処理をステップ S 308 に進め、テーブル定義書作成処理を行う。この処理の詳細は、図 4 を用いて後述する。情報処理装置 100 の CPU 201 は、テーブル定義書作成処理終了後、処理をステップ S 306 に進める。

40

【0050】

情報処理装置 100 の CPU 201 は、ステップ S 306 で、ファイル選択ボタン 703、または、規定フォルダ内の全ての定義ファイルボタン 704 に対する押下指示を受け付けることで入力される DB 操作プログラム作成指示を受け付けたと判定した場合には（ステップ S 309 で YES）、処理をステップ S 310 に進め、DB 操作プログラム作成処理を行う。この処理の詳細は、図 5 を用いて後述する。情報処理装置 100 の CPU 201 は、DB 操作プログラム作成処理終了後、処理をステップ S 306 に進める。

【0051】

50

情報処理装置 100 の CPU 201 は、ステップ S 306 で、ファイル選択ボタン 705、または、規定フォルダ内の全ての定義ファイルボタン 706 に対する押下指示を受け付けることで入力される DDL 作成指示を受け付けたと判定した場合には（ステップ S 311 で YES）、処理をステップ S 312 に進め、DDL 作成処理を行う。この処理の詳細は、図 6 を用いて後述する。情報処理装置 100 の CPU 201 は、DDL 作成処理終了後、処理をステップ S 306 に進める。

【0052】

また、情報処理装置 100 の CPU 201 は、ステップ S 306 で、終了ボタン 707 に対する押下指示を受け付けることで入力される終了指示を受け付けたと判定した場合には（ステップ S 313 で YES）、操作画面 700 の表示を終了し、本図に示す処理を終了する。

10

【0053】

尚、ステップ S 306 で受け付けた操作指示が、上記のいずれでもない場合には（この場合、CPU 201 はステップ S 313 で NO と判定する）、情報処理装置 100 の CPU 201 は処理をステップ S 314 に進め、操作指示に従ったその他の処理を実行する。例えば、データテーブル一覧表示部 701 に表示されているデータテーブルに対するクリック指示を受け付けた場合には、当該データテーブルの選択状態を切り替える（非選択選択、選択 非選択）等の処理を行う。

【0054】

以上が、情報処理装置 100 の CPU 201 によって行われる各種ファイル作成処理の概要の説明である。

20

【0055】

次に、図 4 を参照して、図 3 のステップ S 308 のテーブル定義書作成処理の詳細について説明する。この処理は、情報処理装置 100 の CPU 201 を図 1 に示すテーブル情報入力部 101、テーブル定義書出力部 102 として機能させるプログラムの制御に従って行われる処理である。

【0056】

情報処理装置 100 の CPU 201 は、図 7 の操作画面 700 の作成ボタン 702 に対する押下指示を受け付けると、データテーブル一覧表示部 701 に表示されているデータテーブルのいずれかが選択されているかを判定する（ステップ S 401）。この判定処理で、データテーブルが選択されていない（NO）と判定した場合には、情報処理装置 100 の CPU 201 は処理をステップ S 402 に進め、データテーブルが選択されていない旨のエラーメッセージを出力する。エラーメッセージの出力方法としては、メッセージボックスをディスプレイ装置 210 に表示する方法や、音声警告を行う等の方法を採用することが出来る。ステップ S 402 の処理が終了後、CPU 201 は、本図に示す処理を終了する。

30

【0057】

一方、ステップ S 401 の判定処理で、データテーブルが選択されている（YES）と判定した場合には、選択されている全てのデータテーブルに対して、ステップ S 403 からステップ S 411 の処理を繰り返し実行する。

40

【0058】

まず、ステップ S 403 では、選択されているデータテーブルのうち、まだステップ S 404 以降の処理を行っていないデータテーブルを処理対象のデータテーブルとして特定する。その後、データベース接続情報を用いてデータベースサーバにアクセスし、データベースをオープンする（ステップ S 404）。そして、ステップ S 403 で特定したデータテーブルの列情報（データ項目情報）を取得する（ステップ S 405）。この時、情報処理装置 100 の CPU 201 は、データテーブルの列情報を定義順に昇順に取得することになる。列情報として取得する項目は、それぞれの列（データ項目）の論理名、物理名、型（データ型）、桁数（長さ、精度（小数部））、必須フラグ（登録が必須であるか否かを示す）、主キー番号等である。主キー番号は、主キーの定義順で番号を振ることにな

50

る。例えば、列 A、列 B、列 C が存在するデータテーブルにおいて、主キーが列 C、列 B である場合には、主キー番号として列 C には 1 を、列 B には 2 を付与し、列 A には主キー番号は付与しない。

【 0 0 5 9 】

そして、データテーブルの列情報の取得を終了すると、情報処理装置 1 0 0 は、データベースをクローズする（ステップ S 4 0 6）。

【 0 0 6 0 】

データベースをクローズした後に、情報処理装置 1 0 0 の CPU 2 0 1 は、データベースより取得した各種の情報をを用いて、テーブル定義書の作成処理を行う。まず、情報処理装置 1 0 0 の CPU 2 0 1 は、外部メモリ 2 1 1 に記憶されているテーブル定義書のフォーマットシートをコピーする（ステップ S 4 0 7）。そして、フォーマットシートにヘッダ情報を追加する（ステップ S 4 0 8）。ヘッダ情報とは、例えば、エンティティ名やテーブル名等の情報であり、情報処理装置の CPU 2 0 1 は、これら情報も、データベースから取得する。ヘッダ情報に、スキーマ情報や作成日時情報などの他の情報を追加することも勿論可能である。

【 0 0 6 1 】

ヘッダ情報を追加後、情報処理装置 1 0 0 の CPU 2 0 1 は、ステップ S 4 0 5 で取得した全ての列情報をフォーマットシートに追加する処理を行う。まず、情報処理装置の CPU 2 0 1 は、ステップ S 4 0 5 で取得した列情報のうち、フォーマットシートに追加していない列情報を処理対象とする列情報として特定する（ステップ S 4 0 9）。そして、特定した列情報をフォーマットシートに追加する（ステップ S 4 1 0）。この時、データベースより取得した列の論理名、物理名、型、桁数（長さ、精度（小数部））、必須フラグや主キー番号を、フォーマットシートで規定された箇所に追加することになる。この処理をすべての列情報に対して実行後、フォーマットシートを予め決められたファイル命名規則に従った名称で、テーブル定義書格納フォルダに保存する（ステップ S 4 1 1）。例えば、「テーブル名_拡張子(xls)」などといった命名記憶に従ってファイル名を設定することになる。これにより、テーブル定義書（ファイル）が生成される。

【 0 0 6 2 】

情報処理装置 1 0 0 の CPU 2 0 1 は、ステップ S 4 1 0 の処理終了後、未処理のデータテーブルがあれば、そのデータテーブルについて、上記のステップ S 4 0 3 以降の処理を実行する。未処理のデータテーブルがない場合には、本図に示す処理を終了する。以上が、テーブル定義書作成処理の詳細な処理の説明である。

【 0 0 6 3 】

ここで、図 9 を参照して、図 4 のテーブル定義書作成処理が行われることにより作成されるテーブル定義書の一例について説明する。

【 0 0 6 4 】

図 9 に示すように、テーブル定義書 9 0 0 は、表計算アプリケーションで表示・編集が可能な形式で作成される。テーブル定義書 9 0 0 中の 9 1 0 は、ヘッダ情報エリアであって、エンティティ名 9 1 1 や、テーブル名 9 1 2 等が登録されている。ここに登録されている情報は、図 4 のステップ S 4 0 8 で追加される情報である。

【 0 0 6 5 】

また、9 2 0 は、列情報エリアであって、データベースから取得した列情報が登録される。例えば、列（データ項目）の論理名 9 2 1、物理名 9 2 2、（データ）型 9 2 3、長さ 9 2 4、精度 9 2 5、必須フラグ 9 2 6、主キー番号 9 2 7 等の情報が登録されている。ここに登録されている情報は、図 4 のステップ S 4 1 0 で追加される情報である。以上が、テーブル定義書 9 0 0 の構成の一例の説明である。

【 0 0 6 6 】

尚、このようにデータベースに作成されているデータベースからテーブル定義書を作成する際に、所定のフォーマットデータを用いて、テーブル情報をそのフォーマットデータに追記することによりテーブル定義書を作成することで、テーブル定義書のフォーマット

10

20

30

40

50

を統一させることが可能となる。

【0067】

例えば、データベースシステムを開発するにあたって、複数の開発ベンダーが存在した場合には、それぞれが異なるテーブル定義書のフォーマットを使用している場合もある。そのような場合には、後述するDB操作プログラム作成処理や、DDL作成処理において、複数のテーブル定義書のフォーマットに対応させる必要性が生じるが、本発明では、データベースに既に作成されているデータテーブルから統一したフォーマットのテーブル定義書を作成することが可能なので、複数のフォーマットに対応させる必要がない。

【0068】

次に、図5を参照して、図3のステップS310のDB操作プログラム作成処理の詳細について説明する。このDB操作プログラム処理において、情報処理装置100のCPU201は、図9に示すテーブル定義書を用いて、DB操作プログラム(図11から図13)を作成することになる。この処理は、情報処理装置100のCPU201を図1に示すテーブル定義書入力部103、プログラム出力部104として機能させるプログラムの制御に従って行われる処理である。

10

【0069】

情報処理装置100のCPU201は、まず、DB操作プログラム作成指示が、図7の操作画面700中のファイル選択ボタン703に対しての押下指示により入力されたか、それとも、規定フォルダ内の全ての定義ファイルボタン704に対しての押下指示により入力されたかを判定する(ステップS501)。

20

【0070】

ステップS501の判定処理で、規定フォルダ内の全ての定義ファイルボタン704に対しての押下指示によりDB操作プログラム作成指示が入力された(NO)と判定した場合には、情報処理装置100のCPU201は処理をステップS502に進め、テーブル定義情報格納フォルダ内の全てのテーブル定義書(ファイル)を、DB操作プログラムの作成処理対象に設定する。

【0071】

一方、ファイル選択ボタン703に対しての押下指示によりDB操作プログラム作成指示が入力された(YES)と判定した場合には、情報処理装置100のCPU201は処理をステップS503に進め、テーブル定義情報格納フォルダ内のテーブル定義書の情報(例えばファイル名)を取得し、DB操作プログラムの作成処理対象とするテーブル定義情報を選択するための不図示のファイル選択画面をディスプレイ装置210に表示する(ステップS504)。そして、ファイル選択画面を介して、DB操作プログラム作成処理対象とするテーブル定義書の選択を受け付ける(ステップS505)。そして、情報処理装置100のCPU201は、ステップS505で選択を受け付けたテーブル定義書を、DB操作プログラム作成処理対象に設定する。

30

【0072】

ステップS502、または、ステップS505の処理終了後、DB操作プログラムの作成処理対象に設定した全てのテーブル定義書について、ステップS506からステップS521の処理を行う。

40

【0073】

まず、情報処理装置100のCPU201は、DB操作プログラムの作成処理対象としてテーブル定義書のうち、ステップS507以降の処理をまだ行っていないテーブル定義書を、処理対象のテーブル定義書として特定する(ステップS506)。

【0074】

その後、情報処理装置100のCPU201は、DB操作プログラム出力先フォルダのパス情報を取得する(ステップS507)。そして、取得したパス情報が示すフォルダが作成されているかを判定し(DB操作プログラム出力先フォルダ作成済み?:ステップS508)、作成されていない(NO)と判定した場合には、DB操作プログラム出力先フォルダを作成する(ステップS509)。

50

【0075】

その後、情報処理装置100のCPU201は、DB操作プログラム出力先フォルダに、ステップS506で特定したテーブル定義書を用いて過去に作成されたDB操作プログラムファイルが保存されているか（作成済みであるか）を判定する（ステップS510）。本実施例では、テーブル定義書から作成されるDB操作プログラムは予め決められた規則に従った名称のファイル名で登録されるので、DB操作プログラムのファイル名からそのDB操作プログラムの作成元となったテーブル定義書を特定することが出来る。尚、ファイル名以外に、DB操作プログラム中にコメントとしてテーブル定義書名を記憶することで、作成元のテーブル定義書を特定可能にするように構成しても勿論構わない。

【0076】

ステップS510の判定処理で作成済みである（YES）と判定した場合には、情報処理装置100のCPU201は処理をステップS511に進め、作成済みのDB操作プログラム中のカスタマイズコードを取得し、RAM202に一時保存する。図10に示す通り、本発明で作成されるDB操作プログラム（ソースコード）には、カスタマイズ領域開始特定文字列（図10中では「USER__CODE__START」）1001とカスタマイズ領域終了特定文字列（図10中では「USER__CODE__END」）1002とで特定されるカスタマイズコードの入力箇所であるカスタマイズ領域が設定されており、ユーザは、カスタマイズコードの追加等を行う場合には、そのカスタマイズ領域に追加のソースコードを記入することになる。

【0077】

ステップS510でNOと判定した場合、または、ステップS511の処理終了後、情報処理装置100のCPU201は処理をステップS512に進め、DB操作プログラムのクラス定義部を出力する。ここで、出力されるDB操作プログラムのクラス定義部の構成の一例を図11に示す。

【0078】

図11に示す通り、DB操作プログラムのクラス定義部1100は、当該データ定義書が示すデータベースを操作するためのクラスの情報が定義されたものである。クラス名1101には、データ定義書900のテーブル名912に記載された情報が設定される。また、ここで作成されるクラスは、予め決められた所定の基底クラスを継承した派生クラスとして作成される。そして、コンストラクタ1102等が自動的に作成され、そのソースコードが出力される。

【0079】

図5の説明に戻る。ステップS512の処理終了後、情報処理装置100のCPU201は、テーブル定義書の列情報エリア920に登録されている全ての列情報について、ステップS513、ステップS514の処理を行う。まず、情報処理装置100のCPU201は、ステップS513において、ステップS514の処理をまだ行っていない列情報を処理対象とする列情報と特定する。そして、定数定義部に、当該列情報の名称を管理するための定数を追加する処理を行う（列名称を追加：ステップS514）。ここで、DB操作プログラムの定数定義部の構成の一例を図12に示す。

【0080】

図12に示す通り、DB操作プログラムの定数定義部1200は、テーブル定義書900の列情報エリア920に登録されている各列情報の物理名を文字列として記憶管理するための定数が出力されたものである。

【0081】

本実施例では、物理名に登録された文字列を定数名とした、それぞれの列情報の物理名を管理するための定数を定義している（図中1201）。また、その定数には、物理名の文字列が代入されている（図中1202）。さらに、論理名をコメントとして追加しておくことにより（図中1203）、ソースコード（プログラム）の編集を行うプログラム作成者が、それぞれの定数が、どの列情報を管理しているかをより把握しやすくなるようにしている。

10

20

30

40

50

【 0 0 8 2 】

図5の説明に戻る。テーブル定義書に含まれる全ての列情報に対して、ステップS513、及び、ステップS514の処理が終了後、情報処理装置100のCPU201は、当該テーブル定義書で定義されているデータベースの検索を行うための検索処理部をDB操作プログラムに出力する(ステップS515)。同様に、更新処理実行部、登録処理実行部、作事処理実行部をDB操作プログラムに出力する(それぞれ、ステップS516、ステップS517、ステップS518)。

【 0 0 8 3 】

ステップS515からステップS518で、テーブル定義書で定義されるデータベースの操作を行うための実行部のソースコードを出力した後に、情報処理装置100のCPU201は、カスタイズ領域を特定するための、カスタイズ領域開始特定文字列を出力する(ステップS519)。このカスタイズ領域開始特定文字列は、ソースコードをコンパイル等する際に影響が出ないように、コメントとして出力する。その後、過去に当該テーブル定義書を用いて作成されたDB操作プログラム中のカスタイズコードを、一時保存しているかを判定し(ステップS520)、一時保存している(YES)と判定した場合には、情報処理装置100のCPU201は処理をステップS521に進め、RAM202に一時保存しているカスタイズコードを出力する。

10

【 0 0 8 4 】

ステップS520でNOと判定した場合、または、ステップS521の処理が終了した後に、情報処理装置100のCPU201は処理をステップS522に進め、カスタイズ領域の終了を示すカスタイズ領域終了特定文字列を出力する。このカスタイズ領域終了特定文字列は、ソースコードをコンパイル等する際に影響が出ないように、コメントとして出力する。尚、ステップS520からステップS522の処理を行った結果、ステップS521で出力されたカスタイズコードは、カスタイズ領域開始特定文字列とカスタイズ領域終了特定文字列とに囲まれたカスタイズ領域に出力されることになる。であるために、当該カスタイズコードが削除されない限りは、当該テーブル定義書を用いて再度DB操作プログラムを作成する際にも、再作成されるDB操作プログラムに追加されることになる。

20

【 0 0 8 5 】

図13は、図5のステップS515からステップS522の処理を行うことで出力される処理実行部のプログラム(ソースコード)の一例を示している。

30

【 0 0 8 6 】

図13に示すソースコード中の検索処理実行部に示すソースコードでは、当該データベース単体の検索を行うための機能を提供する。1301は、検索結果で得られる項目は、クラス定数定義部で出力した全ての定数となるようにするためのソースコード、1302は、対象テーブルの列であれば、検索条件として任意に与えられるようにするためのソースコードを出力している。また、1303は、検索結果を主キーに設定された列で昇順にソートされるようにするためのソースコードを示している。

【 0 0 8 7 】

また、図中の1304に示す通り、既に同一のテーブル定義書を用いて自動作成されたプログラムにプログラム開発者により追加されたソースコードが含まれる場合には、カスタイズ領域開始特定文字列1304と、カスタイズ領域終了特定文字列1305とで特定されるカスタイズ領域1306に、過去に追加されたソースコードを出力する。

40

【 0 0 8 8 】

図5の説明に戻る。情報処理装置100のCPU201は、カスタイズ領域終了特定文字列を出力した後に、上記処理により作成されたプログラム(ソースコード)を所定の名称のファイルとして、DB操作プログラム格納先フォルダに保存する(ステップS523)。例えば、「テーブル名+拡張子(.cs)」といった名称のファイルとして保存する。これによりテーブル定義書をもとにしたDB操作プログラムファイルが作成される。

【 0 0 8 9 】

50

そして、情報処理装置100のCPU201は、上記のステップS506からステップS523の処理を、DB操作プログラムの作成処理対象として設定した全てテーブル定義書に対して行った後に、本図に示す処理を終了する。以上が、図3のステップS310のDB操作プログラム作成処理の詳細な処理の説明である。

【0090】

次に、図6を参照して、図3のステップS312のDDL作成処理の詳細について説明する。この処理において、情報処理装置100のCPU201は、図9に示すテーブル定義書を用いて、DDL(図14から図16)を作成することになる。尚、この処理は、情報処理装置100のCPU201を、図1に示すテーブル定義書入力部103、DDL出力部105として機能させるためのプログラムによる制御に従って行われる。

10

【0091】

情報処理装置101のCPU201は、まず、DDL作成指示が、図7の操作画面700中のファイル選択ボタン705に対しての押下指示により入力されたか、それとも、規定フォルダ内のすべての定義ファイルボタン706に対しての押下指示により入力されたかを判定する(ステップS601)。

【0092】

ステップS601の判定処理で、規定フォルダ内の全ての定義ファイルボタン706に対しての押下指示によりDDL作成指示が入力された(NO)と判定した場合には、情報処理装置100のCPU201は処理をステップS602に進め、テーブル定義情報格納フォルダ内の全てのテーブル定義書(ファイル)を、DDLの作成処理対象に設定する。

20

【0093】

一方、ファイル選択ボタン705に対しての押下指示によりDDL作成指示が入力された(YES)と判定した場合には、情報処理装置100のCPU201は処理をステップS603に進め、テーブル定義情報格納フォルダ内のテーブル定義書の情報(例えばファイル名)を取得し、DDLの作成処理対象とするテーブル定義書を選択するための不図示のファイル選択画面をディスプレイ装置210に表示する(ステップS604)。そして、ファイル選択画面を介して、DDLの作成処理対象とするテーブル定義書の選択を受け付ける(ステップS605)。そして、情報処理装置100のCPU201は、ステップS605で選択を受け付けたテーブル定義書をDDL作成処理対象に設定する。

【0094】

30

ステップS602、または、ステップS605の処理終了後、DDL作成処理対象に設定した全てのテーブル定義書について、ステップS606からステップS621の処理を行う。

【0095】

まず、情報処理装置100のCPU201は、DDLの作成処理の対象としたテーブル定義書のうち、ステップS607以降の処理をまだ行っていないテーブル定義書を、処理対象のテーブル定義書として特定する(ステップS606)。

【0096】

その後、情報処理装置100のCPU201は、DDL出力先フォルダのパス情報(ステップS607)を取得する。そして、取得したパス情報が示すフォルダが作成されているかを判定し(ステップS608)、作成されていない(NO)と判定した場合には、DDL出力先フォルダを作成する(ステップS609)。その後、情報処理装置100のCPU201は、DDLの作成を開始する。

40

【0097】

まず、情報処理装置100のCPU201は、テーブル作成対象とするテーブルのCREATE文を出力する。ここでは、「CREATE TABLE テーブル名(」を出力することになる(ステップS610)。尚、この時、CREATE文実行の際にエラーにならないようにするために、CREATE文出力の前に、「DROP TABLE テーブル名;」を出力しておく。

【0098】

50

次に、情報処理装置 100 の CPU 201 は、テーブル定義書 900 の列情報エリア 920 に記載されている各列情報の出力を行う。まず、CPU 201 は、列情報エリア 920 に記載されている列情報を上から順に処理対象の列情報と設定する（ステップ S 6 1 1）。そして、ステップ S 6 1 1 で処理対象に設定した列情報について、「物理名 型 NOT NULL 制約」の形式で出力する（ステップ S 6 1 2）。このステップ S 6 1 1、ステップ S 6 1 2 の処理を、処理対象のテーブル定義書 900 の列情報エリアに記載されている全ての列情報に対して行った後に、情報処理装置 100 の CPU 201 は、CREATE TABLE 文の終了を意味する「);」を出力する（ステップ S 6 1 3）

【0099】

上記のステップ S 6 1 0 からステップ S 6 1 3 の処理を行うことによって、図 14 に示す DDL サンプル 1 が作成される。DDL サンプル 1 は、この DDL の作成に用いたテーブル定義書で定義されたデータテーブルと同様のデータテーブルをデータベース上に作成するためのコードである。

10

【0100】

ステップ S 6 1 3 の処理終了後、情報処理装置 100 の CPU 201 は、テーブルのコメント（論理名）出力処理を行う。この処理では、テーブル定義書 900 の列情報エリア 920 に記載されている全ての列情報の論理名を、「COMMENT ON COLUMN テーブル名.物理名 IS 論理名」の形式で出力する。

【0101】

まず、情報処理装置 100 の CPU 201 は、「COMMENT ON TABLE テーブル名 IS エンティティ名;」を出力する（ステップ S 6 1 4）。その後、テーブル定義書 900 の列情報エリア 920 に記載されている全ての列情報に対して、ステップ S 6 1 5、ステップ S 6 1 6 の処理を行う。

20

【0102】

CPU 201 は、列情報エリア 920 に記載されている列情報を上から順にテーブルコメント出力処理の処理対象の列情報と設定する（ステップ S 6 1 5）。そして、ステップ S 6 1 5 で処理対象に設定した列情報について、「COMMENT ON TABLE テーブル名 IS エンティティ名;」の形式で出力する（ステップ S 6 1 6）。このステップ S 6 1 5、ステップ S 6 1 6 の処理を、処理対象のテーブル定義書 900 の列情報エリアに記載されている全ての列情報に対して行う。

30

【0103】

上記のステップ S 6 1 4 からステップ S 6 1 6 の処理を行うことによって、図 15 に示す DDL サンプル 2 が作成される。この DDL サンプル 2 は、DDL サンプル 1 に示すコードにより作成されるデータテーブルの物理名と論理名の紐付を行うためのコードである。

【0104】

図 6 の説明に戻る。ステップ S 6 1 6 の処理終了後、情報処理装置 100 の CPU 201 は、テーブルの主キーの出力処理を行う。

【0105】

まず、CPU 201 は、主キー制約の CREATE 文を、「ALTER TABLE テーブル名 ADD CONSTRAINT テーブル名_PKEY」で出力する（ステップ S 6 1 7）。尚、この際に、ステップ S 6 1 7 で出力した CREATE 文がエラーにならないように、「ALTER TABLE 物理名 DROP CONSTRAINT テーブル名_PKEY」文をその前に出力しておく。

40

【0106】

その後、テーブル定義書 900 の列情報エリア 920 に記載されている全ての列情報について、ステップ S 6 1 9 からステップ S 6 2 1 の処理を行う。

【0107】

まず、CPU 201 は、列情報エリア 920 に記載されている列情報を上から順に処理対象の列情報と設定する（ステップ S 6 1 9）。その後、処理対象とした列情報の主キー 927 に当該列情報が主キーであることを示す主キー番号が登録されているかを判定する

50

(ステップS620)。ステップS620の判定処理でYESと判定した場合には、情報処理装置100のCPU201は、その列情報の物理名を出力する(ステップS621)。そして、テーブル定義書900の列情報エリアに記載の全ての列情報に対してステップS619からステップS621の処理が終了したかを判定し、まだ終了していないと判定した場合には、処理をステップS619に進め、未処理の列情報に対して、ステップS619からステップS621の処理を行う。そしてすべての列に対してそれら処理を行った後に、終了を示す「」を出力する(ステップS622)

【0108】

上記のステップS619からステップS622の処理を行うことにより、図16に示すDDLサンプル3が作成される。DDLサンプル3は、DDLサンプル1に示すコードにより作成されるデータテーブルに主キー制約を設定するためのコードである。

10

【0109】

ステップS622の処理が終了後、または、ステップS617でNOと判定した場合には、情報処理装置100のCPU201は処理をステップS623に進め、上記処理で出力したDDLを、所定のファイル名で、DDL格納先フォルダに保存する。例えば、「テーブル名+拡張子(.sql)」といった名称のファイルとして保存する。これによりテーブル定義書をもとにしたDDLファイルが作成される。

【0110】

そして、ステップS602またはステップS605でDDLの作成処理対象として設定した全てのテーブル定義書に対するDDL作成処理を終了したと情報処理装置100のCPU201が判定した後に、本図に示す処理を終了する。以上が、図3のステップS312のDDL作成処理の詳細な処理の説明である。

20

【0111】

以上説明したように、本発明によれば、テーブル定義書からDB操作プログラムを作成する場合に、既に、同一のテーブル定義書を用いてプログラムの作成が行われ、その作成されたプログラムに変更が加えられている場合に、その変更を新たに自動生成されるプログラムに反映させた形で、DB操作プログラムを作成することが可能となる。

【0112】

また、既に作成されたデータベースのデータテーブルを用いてテーブル定義書を作成することが可能であるために、異なるテーブル定義書のフォーマットを使用している複数の開発ベンダーにより、データベースシステムが構築された場合であっても、データベースのデータテーブルから統一のフォーマットのテーブル定義書を作成することが可能となる。これにより、DB操作プログラムの作成を行う際に、異なる複数のテーブル定義書のフォーマットに対応させなくても良くなるという効果を生ずる。

30

【0113】

以上、実施形態例を詳述したが、本発明は、方法、プログラムもしくは記録媒体等としての実施態様をとることが可能である。

【0114】

また、本発明におけるプログラムは、各処理方法をコンピュータが実行可能(読み取り可能)なプログラムであり、本発明の記録媒体は、各処理方法をコンピュータが実行可能なプログラムが記憶されている。なお、本発明におけるプログラムは、情報処理装置で行われる処理方法ごとのプログラムであってもよい。

40

【0115】

以上のように、前述した実施形態の機能を実現するプログラムを記録した記録媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ(またはCPUやMPU)が記録媒体に格納されたプログラムを読み取り実行することによっても、本発明の目的が達成されることは言うまでもない。

【0116】

この場合、記録媒体から読み出されたプログラム自体が本発明の新たな機能を実現することになり、そのプログラムを記憶した記録媒体は本発明を構成することになる。

50

【 0 1 1 7 】

プログラムを供給するための記録媒体としては、例えば、フレキシブルディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、DVD-ROM、磁気テープ、不揮発性のメモリカード、ROM、EEPROM、シリコンディスク等を用いることができる。

【 0 1 1 8 】

また、コンピュータが読み出したプログラムを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）等が実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

10

【 0 1 1 9 】

さらに、記録媒体から読み出されたプログラムが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれた後、そのプログラムの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPU等が実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【 0 1 2 0 】

また、本発明は、システムあるいは装置にプログラムを供給することによって達成される場合にも適応できることは言うまでもない。この場合、本発明を達成するためのプログラムを格納した記録媒体を該システムあるいは装置に読み出すことによって、そのシステムあるいは装置が、本発明の効果を享受することが可能となる。

20

【 0 1 2 1 】

さらに、本発明を達成するためのプログラムをネットワーク上のサーバ、データベース等から通信プログラムによりダウンロードして読み出すことによって、そのシステムあるいは装置が、本発明の効果を享受することが可能となる。

【 符号の説明 】

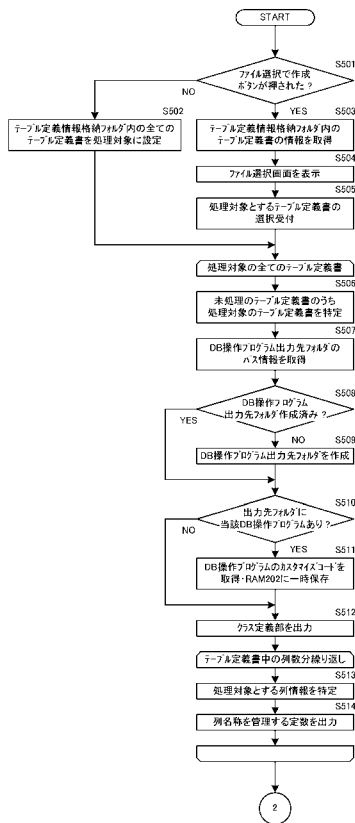
【 0 1 2 2 】

- 1 0 0 情報処理装置
- 2 0 1 CPU
- 2 0 2 RAM
- 2 0 3 ROM
- 2 0 4 システムバス
- 2 0 5 入力コントローラ
- 2 0 6 ビデオコントローラ
- 2 0 7 メモリコントローラ
- 2 0 8 通信I/F（インタフェース）コントローラ
- 2 0 9 入力装置
- 2 1 0 ディスプレイ装置
- 2 1 1 外部メモリ

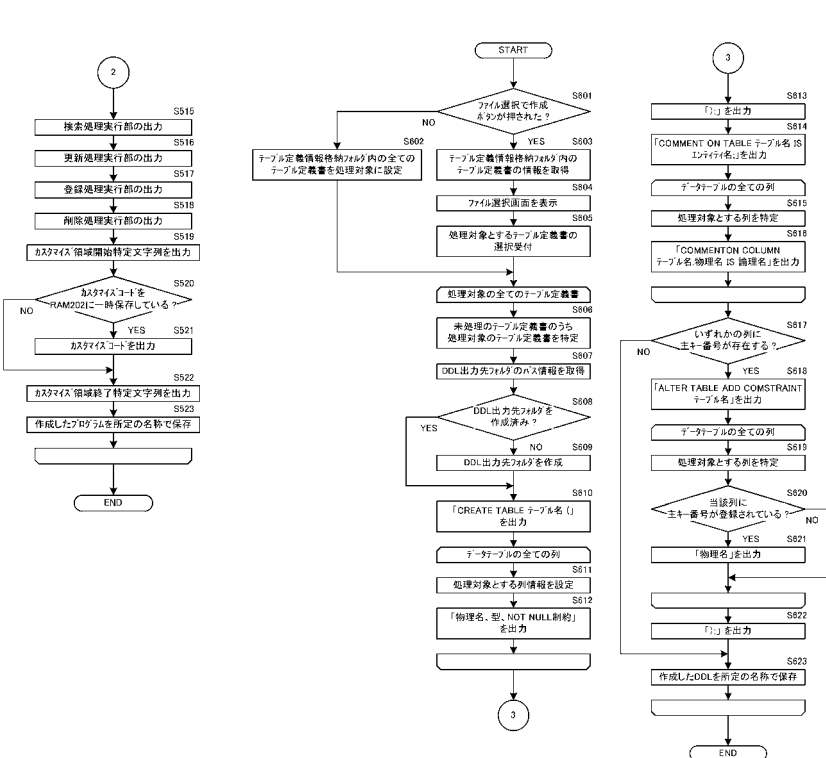
30

40

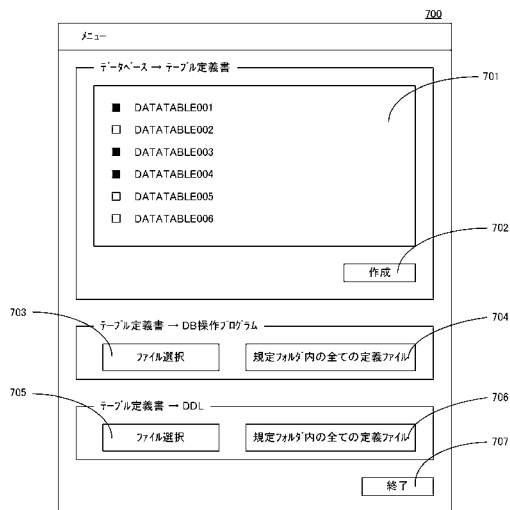
【図5】



【図6】



【図7】



【図8】

| | | | |
|------------------|---------|-----------------|-----|
| データベース接続情報 | DBサーバ | 192.168.1.1 | 801 |
| | ユーザID | administrator | 802 |
| | パスワードID | password | 803 |
| テーブル定義書格納先フォルダ | | C:\Work\TTable | 804 |
| DB操作プログラム格納先フォルダ | | C:\Work\Program | 805 |
| DDL格納先フォルダ | | C:\Work\DDL | 806 |

【 図 9 】

| エンティティ名 | データテーブルID | テーブル名 | DATATABLE001 | | | | |
|---------|-------------|-----------|--------------|-----|-----|-----|-----|
| 論理名 | 物理名 | 型 | 長さ | 精度 | 必須 | 主キー | |
| 1 | データ項目000001 | CD_000001 | VARCHAR2 | 4 | 0 | Y | 1 |
| 2 | データ項目000002 | CD_000002 | VARCHAR2 | 5 | 0 | Y | 2 |
| 3 | データ項目000003 | CD_000003 | VARCHAR2 | 2 | 0 | Y | 3 |
| 4 | データ項目000004 | CD_000004 | VARCHAR2 | 2 | 0 | Y | 4 |
| 5 | データ項目000005 | CD_000005 | VARCHAR2 | 10 | 0 | Y | — |
| 6 | データ項目000006 | CD_000006 | VARCHAR2 | 10 | 0 | Y | — |
| 7 | データ項目000007 | CD_000007 | VARCHAR2 | 20 | 0 | Y | — |
| ... | ... | ... | ... | ... | ... | ... | ... |
| X | データ項目00000X | CD_00000X | VARCHAR2 | 30 | 0 | Y | — |
| Y | データ項目00000Y | CD_00000Y | VARCHAR2 | 8 | 0 | Y | — |
| Z | データ項目00000Z | CD_00000Z | VARCHAR2 | 8 | 0 | Y | — |

【 図 1 0 】

```

...
// USER_CODE_START
#region ソースコードを追加しました。
public Datatable SelectCommand2(Datatable inDatatbl)
{
    StringBuilder sbSQL = new StringBuilder();
    ArrayList alParam = new ArrayList();
    ...
    return base.FillbyDataReader(this.OraCommand, sbSQL.ToString(), apParam);
}
#endregion
// USER_CODE_END
...

```

【 図 1 1 】

```

public partial class DATATABLE001 : BaseDacClass
{
    #region コンストラクタ
    public DATATABLE001()
    {
        #endregion
        ...
    }
}

```

【 図 1 2 】

```

#region フォールド名定義
// <定数> データテーブルID データ項目000001
public const string CD_000001 = "CD_000001"; // データ項目000001
// <定数> データテーブルID データ項目000002
public const string CD_000002 = "CD_000002"; // データ項目000002
// <定数> データテーブルID データ項目000003
public const string CD_000003 = "CD_000003"; // データ項目000003
// <定数> データテーブルID データ項目000004
public const string CD_000004 = "CD_000004"; // データ項目000004
// <定数> データテーブルID データ項目000005
public const string CD_000005 = "CD_000005"; // データ項目000005
// <定数> データテーブルID データ項目000006
public const string CD_000006 = "CD_000006"; // データ項目000006
// <定数> データテーブルID データ項目000007
public const string CD_000007 = "CD_000007"; // データ項目000007
...
// <定数> データテーブルID データ項目00000X
public const string CD_00000X = "CD_00000X"; // データ項目00000X
// <定数> データテーブルID データ項目00000Y
public const string CD_00000Y = "CD_00000Y"; // データ項目00000Y
// <定数> データテーブルID データ項目00000Z
public const string CD_00000Z = "CD_00000Z"; // データ項目00000Z
#endregion

```

【 図 1 3 】

```

~~~~~
#region 検索処理実行部
Private Datable SelectByKey(Datable inDt001, bool flgSearch)
{
    StringBuilder strSql = new StringBuilder();
    strSql.Append("SELECT ");
    strSql.Append("DATatable.cd_000001" + ", ");
    strSql.Append("DATatable.cd_000002" + ", ");
    strSql.Append("DATatable.cd_000003" + ", ");
    :
    strSql.Append("DATatable.cd_00000Z" + ", ");
    strSql.Remove(strSQL.Length - 1, 1);
    strSql.Append(" FROM ");
    strSql.Append(base.tableID);
    strSql.Append(" WHERE " + DATatable.ZZ_DELETE + " = '0' ");
    ArrayList paramArray = new ArrayList();
    int nCnt = 1;
    foreach(DataColumn dc in inDt001.Columns)
    {
        if(dc.DataType.Equals(typeof(ArrayList)))
        {
            :
        }
        else
        {
            :
        }
    }
    strSql.Append(" ORDER BY ");
    strSql.Append(DATatable001.cd_000001 + ", ");
    strSql.Append(DATatable001.cd_000002 + ", ");
    strSql.Append(DATatable001.cd_000003 + ", ");
    strSql.Append(DATatable001.cd_000004 + ", ");
    strSql.Remove(strSQL.Length - 1, 1);
    return base.Select(strSQL.ToString(), paramArray, false);
}
#endregion
更新処理実行部
登録処理実行部
削除処理実行部
// USER_CODE_START
#region フォーストードを追加しました。
public Datable SelectCommand2Name(Datable inDatbl)
{
    StringBuilder sbSQL = new StringBuilder();
    ArrayList apParam = new ArrayList();
    :
    :
    return base.FillByDataReader(this.OracleCommand, sbSQL.ToString(), apParam);
}
#endregion
// USER_CODE_END
}

```

【 図 1 5 】

```

1500
COMMENT ON TABLE DATATABLE_001 IS '予-好-7'4001';
COMMENT ON COLUMN DATATABLE_001.cd_000001 IS '予-9項目000001';
COMMENT ON COLUMN DATATABLE_001.cd_000002 IS '予-9項目000002';
COMMENT ON COLUMN DATATABLE_001.cd_000003 IS '予-9項目000003';
COMMENT ON COLUMN DATATABLE_001.cd_000004 IS '予-9項目000004';
COMMENT ON COLUMN DATATABLE_001.cd_000005 IS '予-9項目000005';
COMMENT ON COLUMN DATATABLE_001.cd_000006 IS '予-9項目000006';
COMMENT ON COLUMN DATATABLE_001.cd_000007 IS '予-9項目000007';
:
:
COMMENT ON COLUMN WHIAT001.cd_00000X IS '予-9項目00000X';
COMMENT ON COLUMN WHIAT001.cd_00000Y IS '予-9項目00000Y';
COMMENT ON COLUMN WHIAT001.cd_00000Z IS '予-9項目00000Z';

```

【 図 1 4 】

```

1400
DROP TABLE DATATABLE_001;
CREATE TABLE DATATABLE_001 (
    cd_000001 VARCHAR2(4) DEFAULT ' ' NOT NULL,
    cd_000002 VARCHAR2(5) DEFAULT ' ' NOT NULL,
    cd_000003 VARCHAR2(2) DEFAULT ' ' NOT NULL,
    cd_000004 VARCHAR2(2) DEFAULT ' ' NOT NULL,
    cd_000005 VARCHAR2(10) DEFAULT ' ' NOT NULL,
    cd_000006 VARCHAR2(10) DEFAULT ' ' NOT NULL,
    cd_000007 VARCHAR2(20) DEFAULT ' ' NOT NULL,
    :
    :
    cd_00000X VARCHAR2(30) DEFAULT ' ' NOT NULL,
    cd_00000Y VARCHAR2(8) DEFAULT ' ' NOT NULL,
    cd_00000Z VARCHAR2(8) DEFAULT ' ' NOT NULL
);

```

【 図 1 6 】

```

1600
ALTER TABLE DATATABLE_001 DROP CONSTRAINT DATATABLE_001_PKEY;
ALTER TABLE DATATABLE_001 ADD CONSTRAINT DATATABLE_001_PKEY PRIMARY KEY(
    cd_000001,
    cd_000002,
    cd_000003,
    cd_000004
);

```

フロントページの続き

(72)発明者 小川 哲也

東京都品川区東品川2丁目4番11号 キヤノンITソリューションズ株式会社内

(72)発明者 福邑 和志

東京都品川区東品川2丁目4番11号 キヤノンITソリューションズ株式会社内

Fターム(参考) 5B376 BC29 BC32 FA21