

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6329552号
(P6329552)

(45) 発行日 平成30年5月23日(2018.5.23)

(24) 登録日 平成30年4月27日(2018.4.27)

(51) Int.Cl.

G06F 12/00 (2006.01)

F 1

G06F 12/00 511Z

請求項の数 8 (全 16 頁)

(21) 出願番号 特願2015-534803 (P2015-534803)
 (86) (22) 出願日 平成25年9月30日 (2013.9.30)
 (65) 公表番号 特表2015-530677 (P2015-530677A)
 (43) 公表日 平成27年10月15日 (2015.10.15)
 (86) 国際出願番号 PCT/US2013/062698
 (87) 国際公開番号 WO2014/055438
 (87) 国際公開日 平成26年4月10日 (2014.4.10)
 審査請求日 平成28年9月26日 (2016.9.26)
 (31) 優先権主張番号 13/633,064
 (32) 優先日 平成24年10月1日 (2012.10.1)
 (33) 優先権主張国 米国(US)

(73) 特許権者 502303739
 オラクル・インターナショナル・コーポレーション
 アメリカ合衆国カリフォルニア州9406
 5レッドウッド・シティー、オラクル・パ
 ークウェイ500
 (74) 代理人 110001195
 特許業務法人深見特許事務所
 (72) 発明者 フアン、アイリーン
 アメリカ合衆国、95070 カリフォル
 ニア州、サラトガ、ドリーン・コート、1
 4150

最終頁に続く

(54) 【発明の名称】 単一テーブルから複数テーブルへの参照データセグメント化

(57) 【特許請求の範囲】

【請求項 1】

マルチテーブルデータセグメント化のためのコンピュータによって実現される方法であつて、前記方法は、

データベーステーブルに関連付けられた複数の定義ファイルの各々をマルチテーブルデータを用いて拡張することと、

複数のマルチテーブルを作成することとを備え、各マルチテーブルは前記複数の定義ファイルのうちの異なる1つに関連付けられており、さらに、

前記データベーステーブルに格納されたデータを前記複数の定義ファイルに基づいて転置することを、備え、前記転置することは、

前記定義ファイルの各々について転置スクリプトを生成することを含み、各転置スクリプトは、その定義ファイルに関連付けられた前記データがその定義ファイルの関連付けられたマルチテーブルと互換性があるようにどのようにセグメント化および変換されるのかを定義し、さらに、

転置した前記データを前記データベーステーブルから前記複数のマルチテーブルに移行させることとを備える、方法。

【請求項 2】

前記マルチテーブルデータは、前記マルチテーブルのうちの1つに関連付けられた識別子と、マルチテーブルフラグとを含む、請求項1に記載のコンピュータによって実現される方法。

【請求項 3】

複数のマルチテーブルを作成することはさらに、

各定義ファイルを解析してその定義ファイルのテーブル情報を求めるこことを含み、前記テーブル情報は、名前、列の数、および列名を含み、さらに、

複数のスクリプトを自動的に生成することを含み、各スクリプトは前記マルチテーブルのうちの1つを定義し、各スクリプトは、前記定義ファイルのうちの異なる1つから解析される前記テーブル情報を使用して生成され、さらに、

前記複数のスクリプトの各々を実行して前記複数のマルチテーブルを作成することを含む、請求項1または2に記載のコンピュータによって実現される方法。

【請求項 4】

10

マッピング定義テーブルを作成することをさらに備え、前記マッピング定義テーブルは、各定義ファイルをその関連付けられたマルチテーブルにマッピングする、請求項1から3のいずれか一項に記載のコンピュータによって実現される方法。

【請求項 5】

転置した前記データを前記データベーステーブルから前記複数のマルチテーブルに移行させることは、

各転置スクリプトを実行して、各定義ファイルに関連付けられた前記データをセグメント化および変換することと、

セグメント化および変換した前記データを、前記データベーステーブルから前記複数のマルチテーブルにコピーすることとを含む、請求項1から4のいずれか一項に記載のコンピュータによって実現される方法。

20

【請求項 6】

転置した前記データが前記複数のマルチテーブルに正常に移行されたことの有効性を確認することをさらに備える、請求項1から5のいずれか一項に記載のコンピュータによって実現される方法。

【請求項 7】

請求項1～6のいずれか一項に記載の方法をコンピュータに実行させる、プログラム。

【請求項 8】

マルチテーブルデータセグメント化のためのシステムであって、

コンピュータ読取可能記憶媒体およびプロセッサを含むコンピュータと、

30

データベーステーブルおよび複数の定義ファイルを含むデータベースと、

前記コンピュータ上で実行するマルチテーブルデータセグメント化ツールとを備え、前記マルチテーブルデータセグメント化ツールは、

前記複数の定義ファイルの各々をマルチテーブルデータを用いて拡張し、

複数のマルチテーブルを作成するように動作可能であり、各マルチテーブルは前記複数の定義ファイルのうちの異なる1つに関連付けられており、さらに、

前記データベーステーブルに格納されたデータを前記複数の定義ファイルに基づいて転置し、前記マルチテーブルデータセグメント化ツールは、前記転置するとき、

前記定義ファイルの各々について転置スクリプトを生成し、各転置スクリプトは、その定義ファイルに関連付けられた前記データがその定義ファイルの関連付けられたマルチテーブルと互換性があるようにどのようにセグメント化および変換されるのかを定義し、さらに、

40

転置した前記データを前記データベーステーブルから前記複数のマルチテーブルに移行するように動作可能である、システム。

【発明の詳細な説明】**【技術分野】****【0001】**

発明の背景

本発明の実施形態は概して、単一テーブルから複数テーブルへの参照データセグメント化を提供するための方法およびシステムに関し、より特定的には、1つのテーブルから複

50

数のテーブルに相互参照データを自動的に移行させることによって相互参照ルックアップパフォーマンスを向上させることに関する。

【背景技術】

【0002】

現在、企業は異なるビジネスニーズに対応するために多くの異なるソフトウェアアプリケーションを利用している。たとえば、ある企業は、顧客対話、アカウント管理、製品流通等に異なるアプリケーションを使用し得る。各アプリケーションは独自の特有のデータモデルを使用する可能性が高く、つまり、1つのアプリケーションのデータは別のアプリケーションのデータとは異なってフォーマット化または編成され得る。これによって、アプリケーションデータが両アプリケーションにおいて同一のエンティティを表わす場合であっても、アプリケーション同士の間で当該アプリケーションデータを直接共有することが困難または不可能になり得る。

10

【0003】

アプリケーション統合は、アプリケーションがデータを共有できるようにし得る処理であり、アプリケーションごとにデータを繰返し手動で再入力することによってもたらされる負担およびエラーの可能性を減少させる。アプリケーション統合を実現する1つの方法は、アプリケーションごとに識別子（ID）をリンクする相互参照（XREF）データベースを共通のフォーマットで使用することである。データがアプリケーション同士の間で受渡されると、当該データは、XREFデータベースに格納された情報を使用して変換され得る。たとえば、第1のアプリケーションが第2のアプリケーションにデータを送信すると、当該データは第1のアプリケーションに関連付けられた第1のフォーマットから共通のフォーマットに変換され得、その後、共通のフォーマットから第2のアプリケーションに関連付けられた第2のフォーマットに変換され得る。

20

【0004】

XREFデータは典型的に、単一のデータベーステーブルに格納される。このテーブルは、何億もの行を含んで非常に大きく成長し得る。テーブルのサイズが大きくなるにつれ、テーブル内のXREFデータをルックアップするのに必要な時間が増加し、パフォーマンスのボトルネックにつながり得る。

【発明の概要】

【0005】

30

発明の簡単な概要

本発明の実施形態は、1つのテーブルから複数のテーブルに相互参照データを自動的に移行させることによって相互参照ルックアップパフォーマンスを向上させる方法およびシステムに関する。ある実施形態に従うと、マルチテーブルデータセグメント化の方法は、データベーステーブルに関連付けられた複数の定義ファイルの各々をマルチテーブルデータを用いて拡張すること（augment）を含み得る。上記方法はさらに、複数のマルチテーブルを作成することを含み得る。各マルチテーブルは、上記複数の定義ファイルのうちの異なる1つに関連付けられている。上記方法はさらに、上記データベーステーブルに格納されたデータを上記複数の定義ファイルに基づいて転置することと、転置した上記データを上記データベーステーブルから上記複数のマルチテーブルに移行させることとを含み得る。いくつかの実施形態では、上記マルチテーブルデータは、上記マルチテーブルのうちの1つに関連付けられた識別子と、マルチテーブルフラグとを含み得る。

40

【0006】

いくつかの実施形態では、上記複数のマルチテーブルは、各定義ファイルを解析して（parse）その定義ファイルのテーブル情報を求め、複数のスクリプトを自動的に生成することによって作成され得、各スクリプトは上記マルチテーブルのうちの1つを定義し、各スクリプトは、上記定義ファイルのうちの異なる1つから解析される上記テーブル情報を使用して生成され、上記複数のマルチテーブルはさらに、上記複数のスクリプトの各々を実行して上記複数のマルチテーブルを作成することによって作成され得る。さらに、各定義ファイルをその関連付けられたマルチテーブルにマッピングするマッピング定義テーブ

50

ルが作成され得る。

【0007】

いくつかの実施形態では、上記データベーステーブルに格納されたデータを転置することは、上記定義ファイルの各々について転置スクリプトを生成することを含み得る。各転置スクリプトは、その定義ファイルに関連付けられた上記データがその定義ファイルの関連付けられたマルチテーブルと互換性があるようにどのようにセグメント化および変換されるのかを定義し得る。さらに、転置した上記データを上記データベーステーブルから上記複数のマルチテーブルに移行させることは、各転置スクリプトを実行して、各定義ファイルに関連付けられた上記データをセグメント化および変換することを含み得る。上記データは、セグメント化および変換されると、上記データベーステーブルから上記複数のマルチテーブルにコピーされ得る。さらに、移行が完了した後、移行した上記データの有効性が確認され得る。

【図面の簡単な説明】

【0008】

【図1】本発明のある実施形態に従う、例示的な相互参照テーブルの抜粋を示す図である。

【図2】本発明のある実施形態に従う、マルチデータセグメント化を実行するためのシステムを示す図である。

【図3】本発明のある実施形態に従う、マルチデータセグメント化を実行する方法を示す図である。

【図4A】本発明のある実施形態に従う、例示的な移行されたマルチテーブルを示す図である。

【図4B】本発明のある実施形態に従う、例示的な移行されたマルチテーブルを示す図である。

【図4C】本発明のある実施形態に従う、例示的な移行されたマルチテーブルを示す図である。

【図5】本発明のある実施形態に従う、例示的なモニターテーブルを示す図である。

【図6】本発明のいくつかの実施形態に従って使用され得るシステム環境の構成要素を示す簡略ブロック図である。

【図7】本発明の実施形態に従って使用され得る演算システムの簡略ブロック図である。

【発明を実施するための形態】

【0009】

発明の詳細な説明

以下の説明では、本発明の実施形態が十分に理解されるようにするために、説明の目的で具体的な詳細を記載する。しかし、これらの具体的な詳細がなくても本発明の実施形態を実施できることは明らかであろう。

【0010】

本発明の実施形態は、マルチテーブルデータセグメント化を実行することによって相互参照ルックアップパフォーマンスを向上させるために使用され得る。ある実施形態に従うと、マルチテーブルデータセグメント化の方法は、データベーステーブルに関連付けられた複数の定義ファイルの各々をマルチテーブルデータを用いて拡張することを含み得る。上記方法はさらに、複数のマルチテーブルを作成することを含み得る。各マルチテーブルは、上記複数の定義ファイルのうちの異なる1つに関連付けられている。上記方法はさらに、上記データベーステーブルに格納されたデータを上記複数の定義ファイルに基づいて転置することと、転置した上記データを上記データベーステーブルから上記複数のマルチテーブルに移行させることとを含み得る。

【0011】

図1は、本発明の例示的な相互参照テーブル100の抜粋を示す。図1に示されるように、相互参照テーブルは、異なるアプリケーションにおいて作成される複数の異なるエンティティの相互参照(XREF)データを含み得る。相互参照テーブルは、これらのエン

10

20

30

40

50

ティティのアイデンティティを相関付けるために使用され得、ビジネスによって利用されるアプリケーションおよびエンティティの数が増加するにつれて非常に大きく成長し得る。相互参照テーブルは、X R E F テーブル名 1 0 2 および行番号 1 0 4 によって編成され得る。X R E F テーブル名 1 0 2 は、相互参照テーブルを含む複数の構成仮想テーブルに対応する。たとえば、図 1 において、示される相互参照テーブルの部分は B u s i n e s s _ U n i t _ I D X R E F エンティティに対応し、したがって、X R E F テーブル名 1 0 2 列の下で、B u s i n e s s _ U n i t _ I D X R E F アーティファクト 1 0 6 についての経路がリストされている。各 X R E F アーティファクトは、以下にさらに説明するように、その X R E F アーティファクトに関連付けられた X R E F エンティティの、相互参照テーブル内の他のエンティティへのマッピングを定義する X M L ファイルであり得る。1 0 8 に示されるように、図示されない追加的な行は、異なる X R E F エンティティについての異なる仮想テーブルに対応し得る。 10

【 0 0 1 2 】

各行番号 1 0 4 は、その行の X R E F _ T a b l e _ N a m e に対応する X R E F エンティティの異なるインスタンスを表わす。ゆえに、行番号 3, 4 および 5 の各々は、B u s i n e s s _ U n i t _ I D の異なるインスタンスに対応する。たとえば、図 1 に示されるように、2 つのアプリケーション識別子 (I D) と、B u s i n e s s _ U n i t _ I D X R E F エンティティの各インスタンスについての共通値 (E B I Z _ 0 1 1 1 0 ; J D E E 1 _ 0 1 1 1 2 ; および C O M M O N 1 1 4) との間にマッピングが存在する。したがって、たとえば、行番号 3 に対応するインスタンスに関して、E B I Z _ 0 1 _ 0 3 、J D E E 1 _ 0 1 _ 0 3 および C o m m o n _ 0 3 の各々が互いにマッピングする。 20

【 0 0 1 3 】

図 2 は、本発明のある実施形態に従う、マルチデータセグメント化を実行するためのシステムを示す。図 2 に示されるように、そのようなシステムは、サーバ 2 0 0 と、サーバにアクセス可能な相互参照データベース 2 0 2 とを含み得る。ある実施形態に従うと、サーバは、ローカルエリアネットワーク上で、またはインターネット上で、データベースと直接通信し得る。サーバは、コンピュータ読取可能媒体およびプロセッサを含むコンピュータ、またはコンピュータのクラスタであり得る。サーバは、自身の上で実行するマルチテーブルデータセグメント化ツール 2 0 4 を含み得る。相互参照データベース 2 0 2 は、図 1 に関して上記に説明したような相互参照データを含む X R E F _ D A T A テーブル 2 0 6 を含み得る。マルチテーブルデータセグメント化ツール 2 0 4 を使用して、X R E F _ D A T A テーブル 2 0 6 に格納された相互参照データはセグメント化され、複数のマルチテーブル 2 0 8 に移行され得る。この処理は、以下により詳細に説明する。 30

【 0 0 1 4 】

図 3 は、本発明のある実施形態に従う、マルチデータセグメント化を実行する方法を示す。ある実施形態に従うと、以下に説明する方法は、図 2 に関して上記に説明したマルチテーブルデータセグメント化ツールによって実行され得る。単一の相互参照テーブルが成長するにつれ、参照をルックアップするのに費やす時間が増加してパフォーマンスボトルネックにつながり得る。そのような場合、単一の相互参照テーブルを、各々が異なる X R E F アーティファクトに対応する複数のより小さいテーブルに分割することが有利であり得る。これらのより小さいテーブルの各々は本明細書ではマルチテーブルと称され、相互参照テーブルを複数のマルチテーブルに分割する処理は本明細書ではマルチデータセグメント化と称される。 40

【 0 0 1 5 】

図 3 に示されるように、マルチデータセグメント化は、相互参照データが正確に移行され、必要に応じて迅速に識別および使用され得ることを確実にするための複数のステップを含み得る。本発明のある実施形態に従うと、マルチデータセグメント化処理は、単一の相互参照テーブルから複数の相互参照マルチテーブルにデータを生成し、組織化し、転置し、移行させ、有効性を確認するために提供される。 50

【0016】

ステップ300において、新たな相互参照定義がマルチテーブルについて生成される。これらの定義は、作成すべきマルチテーブルデータベーステーブルの構造を定義するために使用され得る。ステップ302において、新たな相互参照定義を生成することは、データベースをスキャンして元の相互参照定義ファイル（本明細書ではXREFアーティファクトとも称される）を識別してから、この情報を使用して新たな相互参照定義を導出することを含み得る。ステップ304において、新たな定義ファイルの各々はマルチテーブルデータを含むように拡張され得、SQLスクリプト等のスクリプトが拡張した定義ファイルに基づいて生成され得る。これらのスクリプトは実行されると、拡張した定義ファイルに対応する新たなマルチテーブルを作成し得る。このマルチテーブルデータは、XREFデータがマルチテーブルに移動したことを示すフラグと、XREFデータが移動した先のマルチテーブルの識別子とを含む付加的なメタデータであり得る。たとえば、以下に示されるように、`<table>`ノードの下には、この例では値「`specific`」を使用することによって、XREFデータがマルチテーブルに移動したことを示すフラグとして「`mode`」属性が使用され得る。「`dbtable`」属性は、データベース内に作成されることになる対応のマルチテーブルの名前等のマルチテーブルの識別子を含み得る。

【0017】

【数1】

```

<?xml version = '1.0' encoding = 'UTF-8'?>
<!-- Upgraded by Xref Upgrade Utility 1.0 -->
<xref xmlns="http://xmlns.oracle.com/xref">
  <table name="BUSINESS_UNIT_ID" dbtable="XREF_BUSINESS_UNIT_ID"
 mode="specific">
    <description>
    </description>
    <columns>
      <column name="COMMON"/>
      <column name="EBIZ_01"/>
      <column name="JDEE1_01"/>
    </columns>
  </table>
</xref>

```

10

20

30

リストイング1. メタデータが更新されたXREFアーティファクト

【0018】

ステップ306において、新たな相互参照定義ファイルの生成後、新たなマルチテーブルデータベーステーブルが識別および定義され得る。ある実施形態に従うと、ステップ308において、新たなマルチテーブルは、新たな相互参照定義ファイルごとに1つの新たな対応するマルチテーブルを作成することによってターゲットにされる。新たな相互参照定義ファイルの各々は解析され、テーブル名、列名、列の数等、新たなマルチテーブルを作成する際に使用する適切な情報が求められ得る。各テーブルは、新たな相互参照定義ファイルの各々において「`dbtable`」メタデータに従って名前を付けられ得る。列の数および列名などの各マルチテーブルの詳細は、新たな相互参照定義ファイルの各々における`<column>`要素から1対1の対応で求められ得る。

【0019】

40

【数2】

```
CREATE TABLE XREF_BUSINESS_UNIT_ID(
  ROW_ID VARCHAR2(48) NOT NULL,
  COMMON VARCHAR2(2000),
  EBIZ_01 VARCHAR2(2000),
  JDEE1_01 VARCHAR2(2000),
  LAST_MODIFIED TIMESTAMP NOT NULL
);
```

リストイング2. BUSINESS_UNIT_IDの新たな相互参照

10

アーティファクトについてのサンプルマルチテーブルスキーマ

【0020】

上記のリストイング2に示されるように、新たなマルチテーブルの各行は、新たな相互参照定義から導出される列を含み得る。さらに、ある実施形態に従うと、2つの付加的な列ROW_IDおよびLAST_MODIFIEDが追加され得る。ある実施形態に従うと、データベーステーブル名または他の識別子は限られた長さを有し得るため、ステップ310において、定義ファイル内のマルチテーブル識別子をその対応する実際のデータベーステーブル識別子にリンクするためのマッピングテーブルが作成され得る。ある実施形態に従うと、新たなマルチテーブルデータベーステーブルの各々は、1つ以上のSQLスクリプトを使用して作成され得る。

20

【0021】

ステップ312において、XREFデータは転置される。転置ステップの間、ステップ314において、データをマルチテーブル上にセグメント化するための実行プランが1つ以上のSQLスクリプトの形態で生成され、新たな相互参照アーティファクトごとに1つのスクリプトが作成される。転置は、所与の相互参照定義についてのXREFデータが完全にマッピングされるか否かに依存して異なって実行され得る。データが完全にマッピングされる元のテーブル内の相互参照アーティファクトの各々について、アーティファクトの列は、行番号ごとに定義される値を有する。これらの完全にマッピングされるテーブルを移行させるために、データは、対応するマルチテーブルのフォーマットにバルクで転置され得る。

30

【0022】

完全にマッピングされないマルチテーブルは、ナル値および/または1対多マッピングを含み得る。マルチテーブル内のナル値は、いくつかの相互参照アーティファクトの列についてのマッピング値が欠けている元の单一の相互参照テーブルに起因する。この場合、完全にマッピングされるテーブルと同様に、データは移行用に転置され得る。1対多マッピングを有するテーブルについては、1つのアプリケーションについてのIDが別のアプリケーションについての複数のIDにマッピングされ、すなわち、一列内のある値はテーブルの別の列内のいくつかの値に対応し得る。1対多マッピングを有するアーティファクトは、自身のデータを单一の相互参照テーブルから一度に一行ずつ移行させ得る。

40

【0023】

ステップ316において、XREFデータは、ステップ314で生成されたスクリプトを実行することによって单一の相互参照テーブルから対応するマルチテーブルの各々に移行される。各スクリプトは、新たな相互参照アーティファクトの各々についてのデータを、その新たな相互参照アーティファクトのマルチテーブルと互換性のある形態にセグメント化および変換した後、セグメント化および変換したデータを新たなマルチテーブルにコピーする。318に示されるように、各スクリプトは並列実行用に設計されてパフォーマンスを向上させ得る。

【0024】

図4A、図4Bおよび図4Cは、本発明のある実施形態に従う、例示的な移行されたマ

50

ルチテーブルを示す。図4 Aは、ある実施形態に従う、`XREF_BUSINESS_UNI`ターケティングアクトについての例示的な移行されて完全にマッピングされるテーブルを示す。図4 Aに示されるように、各行ID 400は図1の行番号104に対応する。同様に、`EBIZ_01_110`、`JDEE1_01_112`および`COMMON114`について、リストイング2に示されるスキーマに従って列が作成されている。さらに、行が最後に修正された日付を含む新たな`Last Modified`列402が追加されている。

【0025】

図4 Bは、相互参照アーケティングアクトがナル値を含む例を示す。列がその相互参照定義におけるエンティティについて定義されるがその列に値が割当てられない場合は、当該列は`XREF_DATA`テーブルに示されない。ナル値を含む列が組織化ステップ306において識別され、関連付けられたマルチテーブルに含められる。図4 Bに示されるように、マルチテーブル404に関連付けられたエンティティは、3つのアプリケーションID、すなわち`EBIZ_01_406`、`AGILE_01_408`および`PLM4P_410`についての列を含む。`PLM4P`には値が割当てられていないため、各エントリは(ナル)412として示される。

【0026】

図4 Cは、相互参照アーケティングアクトが1対多マッピングを含む例を示す。1対多マッピングを含む移行されたアーケティングアクトは、1つ以上の列内の単一値が異なる列内の多値にマッピングされる、結果として得られるマルチテーブルを有する。図4 Cに示されるように、別個の行IDごとに1つの「完全な」行があり、他のマッピングされた値は、行IDおよび他の列についてのナル値のみを有して各自の行内にある。図4 Cの例では、`OTM_01`値414の各々が3つの`EBIZ_01`値416にマッピングする(たとえば、`OTM_01_1028`が`EBIZ_01_1028A`、`EBIZ_01_1028B`および`EBIZ_01_1028C`にマッピングする)。結果として得られるマルチテーブルでは、同じ行ID(たとえば1028)を共有する3行が組合わさせて、マッピングを完全に定義する1つの「完全な」行を形成する。ある実施形態に従うと、各列内のデータを複数の行について複製する代わりに、418に示されるように(ナル)がテーブルに挿入されてもよい。

【0027】

図3に戻って、ステップ320において、移行の有効性が確認される。有効性の確認は、セグメント化処理および結果の完全性を検証し得る。移行の間、モニタリング統計が収集されてモニターテーブルに格納される。これによって移行を検証し、相互参照アーケティングアクトごとにデータセグメント化処理がエラーなしで完了したことを確かめることができる。

【0028】

図5は、本発明のある実施形態に従う、例示的なモニターテーブルを示す。すべてのアーケティングアクトが確実に移行されるようにするために、データセグメント化の最後に結果確認SQLスクリプトを実行して、マルチテーブルに関連付けられたメトリクスを`XREF_DATA`テーブルに関連付けられたメトリクスと比較することができる。図5に示されるように、モニターテーブル500は、各`XREF`エンティティおよびそのエンティティのデータの移行に関する情報を格納し得る。これは、`XREF`アーケティングアクト名502と、元の`XREF_DATA`テーブル内のこのアーケティングアクトに関連付けられた行の数504とを含み得る。ある実施形態に従うと、この行の数504は、上述のように独立して移行され得る1対多の行を含まない。モニターテーブルは、移行された行の数506を追跡するために使用されてもよい。このように、移行はバッチで起こることができ、エラーが起こった場合に時間および処理要件を節約する。たとえば、百万行を移行させる場合に、エラーが起こって百万行すべてのロールバックが必要になると、これはシステムに多大な処理負担を課す。しかし、移行の進行を追跡し、たとえば1万行ずつバッチで進行することによって、はるかに小さいロールバックでエラーに対処することができる。1対多の行

10

20

30

40

50

は、XREF__DATA内の1対多の行の数508および移行された1対多の行の数510を格納することによって「正規の」行と同様に別個に追跡され得る。さらに、処理が開始する時間512、1対多の行の識別が開始した時間514、および1対多の行の識別の合計時間516を含むタイムスタンプが移行について記録され得る。また、「正規の」行の処理が開始した時間518および正規の行の処理の合計時間520、ならびに同様に、1対多の行の処理が開始した時間522および1対多の行の処理の合計時間524も記録される。エラーが起こると、現在の処理バッチをロールバックして再実行する必要があることを示す例外フラグ526が格納され得る。ある実施形態に従うと、移行の有効性を確認するために、移行されたアーティファクトの数を元のXREF__DATAテーブル内の一意の相互参照定義の数と比較する1つ以上の有効性確認スクリプトが実行され得、有効であるためには、これらの数は等しくなくてはならない。また、元のテーブル内の行の数が移行された行の数と比較され得、有効であるためには、これらの数も等しくなくてはならない。10

【0029】

いくつかの実施形態において、マルチテーブルデータセグメント化のための装置は、データベーステーブルに関連付けられた複数の定義ファイルの各々をマルチテーブルデータを用いて拡張するための手段と、複数のマルチテーブルを作成するための手段とを含み、各マルチテーブルは上記複数の定義ファイルのうちの異なる1つに関連付けられており、さらに、上記データベーステーブルに格納されたデータを上記複数の定義ファイルに基づいて転置するための手段と、転置した上記データを上記データベーステーブルから上記複数のマルチテーブルに移行させるための手段とを含む。20

【0030】

いくつかの実施形態において、上記マルチテーブルデータは、上記マルチテーブルのうちの1つに関連付けられた識別子と、マルチテーブルフラグとを含む。

【0031】

いくつかの実施形態において、上記複数のマルチテーブルを作成するための手段はさらに、各定義ファイルを解析してその定義ファイルのテーブル情報を求めるための手段を含み、上記テーブル情報は、名前、列の数、および列名を含み、上記複数のマルチテーブルを作成するための手段はさらに、複数のスクリプトを自動的に生成するための手段を含み、各スクリプトは上記マルチテーブルのうちの1つを定義し、各スクリプトは、上記定義ファイルのうちの異なる1つから解析される上記テーブル情報を使用して生成され、上記複数のマルチテーブルを作成するための手段はさらに、上記複数のスクリプトの各々を実行して上記複数のマルチテーブルを作成するための手段を含む。30

【0032】

いくつかの実施形態において、上記装置はさらに、マッピング定義テーブルを作成するための手段を含み、上記マッピング定義テーブルは、各定義ファイルをその関連付けられたマルチテーブルにマッピングする。

【0033】

いくつかの実施形態において、上記データベーステーブルに格納されたデータを上記複数の定義ファイルに基づいて転置するための手段はさらに、上記定義ファイルの各々について転置スクリプトを生成するための手段を含み、各転置スクリプトは、その定義ファイルに関連付けられた上記データがその定義ファイルの関連付けられたマルチテーブルと互換性があるようにどのようにセグメント化および変換されるのかを定義する。40

【0034】

いくつかの実施形態において、転置した上記データを上記データベーステーブルから上記複数のマルチテーブルに移行させるための手段はさらに、各転置スクリプトを実行して、各定義ファイルに関連付けられた上記データをセグメント化および変換するための手段と、セグメント化および変換した上記データを、上記データベーステーブルから上記複数のマルチテーブルにコピーするための手段とを含む。

【0035】

50

いくつかの実施形態において、上記装置はさらに、転置した上記データが上記複数のマルチテーブルに正常に移行されたことの有効性を確認するための手段を含む。

【0036】

いくつかの実施形態において、マルチテーブルデータセグメント化のためのシステムは、複数の定義ファイルを含むデータベースと、データベースにアクセス可能に接続され、マルチテーブルデータセグメント化ツールを含むサーバと含み、上記マルチテーブルデータセグメント化ツールは、上記複数の定義ファイルの各々をマルチテーブルデータを用いて拡張し、複数のマルチテーブルを作成するように動作可能であり、各マルチテーブルは上記複数の定義ファイルのうちの異なる1つに関連付けられており、上記マルチテーブルデータセグメント化ツールはさらに、上記データベーステーブルに格納されたデータを上記複数の定義ファイルに基づいて転置し、転置した上記データを上記データベーステーブルから上記複数のマルチテーブルに移行させるように動作可能である。

10

【0037】

いくつかの実施形態において、上記マルチテーブルデータセグメント化ツールは複数のマルチテーブルを作成し、各マルチテーブルは上記複数の定義ファイルのうちの異なる1つに関連付けられており、上記マルチテーブルデータセグメント化ツールはさらに、各定義ファイルを解析してその定義ファイルのテーブル情報を求めるように動作可能であり、上記テーブル情報は、名前、列の数、および列名を含み、上記マルチテーブルデータセグメント化ツールはさらに、複数のスクリプトを自動的に生成するように動作可能であり、各スクリプトは上記マルチテーブルのうちの1つを定義し、各スクリプトは、上記定義ファイルのうちの異なる1つから解析される上記テーブル情報を使用して生成され、上記マルチテーブルデータセグメント化ツールはさらに、上記複数のスクリプトの各々を実行して上記複数のマルチテーブルを作成するように動作可能である。

20

【0038】

いくつかの実施形態において、マルチテーブルデータセグメント化ツールはさらに、マッピング定義テーブルを生成するように動作可能であり、上記マッピング定義テーブルは、各定義ファイルをその関連付けられたマルチテーブルにマッピングし、上記マルチテーブルデータセグメント化ツールはさらに、上記移行についてのデータを格納するモニタテーブルを生成するように動作可能であり、上記モニタテーブルは、上記移行の有効性を確認するために上記マルチテーブルデータセグメント化ツールによって使用され得る。

30

【0039】

いくつかの実施形態において、上記マルチテーブルデータセグメント化ツールは、上記データベーステーブルに格納されたデータを上記複数の定義ファイルに基づいて転置し、上記マルチテーブルデータセグメント化ツールはさらに、上記定義ファイルの各々について転置スクリプトを生成するように動作可能であり、各転置スクリプトは、その定義ファイルに関連付けられた上記データがその定義ファイルの関連付けられたマルチテーブルと互換性があるようにどのようにセグメント化および変換されるのかを定義する。

【0040】

いくつかの実施形態において、上記マルチテーブルデータセグメント化ツールが、転置した上記データを上記データベーステーブルから上記複数のマルチテーブルに移行させると、上記マルチテーブルデータセグメント化ツールはさらに、各転置スクリプトを実行して、各定義ファイルに関連付けられた上記データをセグメント化および変換し、セグメント化および変換した上記データを、上記データベーステーブルから上記複数のマルチテーブルにコピーするように動作可能である。

40

【0041】

図6は、本発明のいくつかの実施形態に従って使用され得るシステム環境600の構成要素を示す簡略ブロック図である。示されるように、システム環境600は、ウェブブラウザ、プロプライエタリクライアント(たとえばOracle Forms)等のクライアントアプリケーションを動作させるように構成される1つ以上のクライアント演算装置602, 604, 606, 608を含む。さまざまな実施形態において、クライアント演算装置602

50

, 604, 606 および 608 はサーバ 612 と対話し得る。

【0042】

クライアント演算装置 602, 604, 606, 608 は、汎用パーソナルコンピュータ（一例として、マイクロソフトウィンドウズ（登録商標）および / またはアップルマックintosh オペレーティングシステムのさまざまなバージョンを実行するパーソナルコンピュータおよび / またはラップトップコンピュータを含む）、（マイクロソフトウィンドウズモバイルなどのソフトウェアを実行し、インターネット、電子メール、SMS、ブラックベリー、または使用可能な他の通信プロトコルである）携帯電話もしくは PDA、および / またはさまざまな市販の UNIX（登録商標）もしくは UNIX のようなオペレーティングシステム（さまざまな GNU / Linux（登録商標）オペレーティングシステムを含むがこれに限定されるものではない）のいずれかを実行するワークステーションコンピュータであってもよい。代替的に、クライアント演算装置 602, 604, 606 および 608 は、ネットワーク（たとえば以下に説明するネットワーク 610）を介して通信することができるシン・クライアントコンピュータ、インターネットへの接続が可能なゲーム機および / またはパーソナルメッセージング装置などのその他の電子装置であってもよい。システム環境 600 は 4 つのクライアント演算装置とともに示されているが、いかなる数のクライアント演算装置がサポートされてもよい。センサを有する装置などの他の装置がサーバ 612 と対話してもよい。1 つ以上のクライアント装置から認可要求を受信してもよい。

【0043】

システム環境 600 はネットワーク 610 を含み得る。ネットワーク 610 は、TCP / IP、SNA、IPX、AppleTalkなどを含むがこれらに限定されるものではないさまざまな市販のプロトコルのいずれかを使用してデータ通信をサポートし得る、当業者になじみのある任意のタイプのネットワークであってもよい。単に一例として、ネットワーク 610 は、イーサネット（登録商標）ネットワーク、トーカン・リング・ネットワークなどのローカルエリアネットワーク（local area network : LAN）、広域ネットワーク、仮想プライベートネットワーク（virtual private network : VPN）を含むがこれに限定されるものではない仮想ネットワーク、インターネット、インターネット、エクストラネット、公衆交換電話網（public switched telephone network : PSTN）、赤外線ネットワーク、無線ネットワーク（たとえば IEEE 802.11 の一連のプロトコル、当該技術分野において公知のブルートゥース（登録商標）プロトコルおよび / またはその他の無線プロトコルのいずれかの下で動作するネットワーク）、ならびに / または、これらのおよび / もしくは他のネットワークの任意の組み合わせであってもよい。

【0044】

システム環境 600 は、汎用コンピュータ、特化サーバコンピュータ（一例として、PC サーバ、UNIX サーバ、ミッドレンジサーバ、メインフレームコンピュータ、ラックマウント式のサーバ等を含む）、サーバファーム、サーバクラスタ、またはその他の適切な配置および / もしくは組み合わせであり得る 1 つ以上のサーバコンピュータ 612 をさらに含む。さまざまな実施形態において、サーバ 612 は、上記の開示において説明した 1 つ以上のサービスまたはソフトウェアアプリケーションで実行するように適合されてもよい。

【0045】

サーバ 612 は、上述のものを含むオペレーティングシステム、および任意の市販のサーバオペレーティングシステムを実行し得る。サーバ 612 はさらに、HTTP サーバ、FTP サーバ、CGI サーバ、Java（登録商標）サーバ、データベースサーバなどを含む、さまざまな追加のサーバアプリケーションおよび / もしくは中間層アプリケーションのいずれかを実行し得る。例示的なデータベースサーバとしては、オラクル、マイクロソフト、サイバース、IBM などから市販されているものが挙げられるが、これらに限定されるものではない。

【0046】

10

20

30

40

50

システム環境 600 はさらに、1つ以上のデータベース 614, 616 を含んでいてよい。データベース 614, 616 はさまざまな場所に存在してもよい。一例として、1つ以上のデータベース 614, 616 は、サーバ 612 にローカルな（および / またはサーバ 612 内に常駐する）記憶媒体上に存在してもよい。代替的に、データベース 614, 616 はサーバ 612 から離れており、ネットワークによる接続または専用の接続を介してサーバ 612 と通信状態にあってもよい。一組の実施形態では、データベース 614, 616 は、当業者になじみのあるストレージエリアネットワーク (storage-area network : SAN) 内に存在してもよい。同様に、サーバ 612 に帰する機能を実行するための任意の必要なファイルが、サーバ 612 上に局所的に、および / または遠隔に、適宜格納されてもよい。一組の実施形態では、データベース 614, 616 は、SQL フォーマット化されたコマンドおよび他のコマンドに応答してデータを格納、更新、および検索するように適合される、Oracle 10g、11g、Release 12 等のリレーションナルデータベースを含んでいてもよい。一実施形態では、一組のリソースを制御するように構成されるポリシーがデータベースの 1 つによって格納されてもよい。

【0047】

図 7 は、本発明の実施形態に従って使用され得る演算システム 700 の簡略ブロック図である。たとえば、図 2 に示されるシステムは、システム 700 等のシステムを使用して実現され得る。バス 724 を介して電気的に結合され得るハードウェア要素を含むコンピュータシステム 700 が示されている。ハードウェア要素は、1つ以上の中央演算処理装置 (central processing unit : CPU) 702 と、1つ以上の入力装置 704 (たとえばマウス、キーボード等) と、1つ以上の出力装置 706 (たとえばディスプレイ装置、プリンタ等) とを含んでいてもよい。CPU は、シングルコアまたはマルチコア CPU を含んでいてもよい。また、コンピュータシステム 700 は、1つ以上の記憶装置 708 も含んでいてもよい。一例として、記憶装置 708 は、ディスクドライブ、光学記憶装置、およびランダムアクセスメモリ (random access memory : RAM) および / またはリードオンリメモリ (read-only memory : ROM) などの、プログラム可能、フラッシュ更新可能などであり得るソリッドステート記憶装置、といった装置を含んでいてもよい。

【0048】

さらに、コンピュータシステム 700 は、コンピュータ読取可能記憶媒体リーダ 712 と、通信サブシステム 714 (たとえばモデム、ネットワークカード (無線または有線)、赤外線通信装置等) と、上記の RAM および ROM 装置を含み得るワーキングメモリ 718 とを含んでいてもよい。いくつかの実施形態では、コンピュータシステム 700 は、デジタル信号プロセッサ (digital signal processor : DSP)、特殊用途プロセッサなどを含み得る処理加速ユニット 716 も含んでいてもよい。

【0049】

さらに、コンピュータ読取可能記憶媒体リーダ 712 は、ともに (および任意に、記憶装置 708 と組み合わせて) 一時的および / またはより永久的にコンピュータ読取可能な情報を含むためのリモートの、ローカルの、固定された、および / または取外し可能な記憶装置プラス記憶媒体を包括的に表わすコンピュータ読取可能記憶媒体 710 に接続可能である。通信システム 714 は、ネットワークおよび / またはそのシステム環境に関して上記に説明した任意のその他のコンピュータとデータをやり取りすることを可能にし得る。

【0050】

また、コンピュータシステム 700 は、現在のところワーキングメモリ 718 内に位置しているように示されているソフトウェア要素も含んでいてもよく、当該ソフトウェア要素は、オペレーティングシステム 720、および / または、(クライアントアプリケーション、ウェブブラウザ、中間層アプリケーション、RDBMS 等であってもよい) アプリケーションプログラムなどの他のコード 722 を含む。例示的な実施形態では、ワーキングメモリ 718 は、実行可能なコードと、上記の認可要求を処理するために使用されるメモリ構造などの関連付けられたデータ構造とを含んでいてもよい。コンピュータシステム

10

20

30

40

50

700の代替的な実施形態は、上記のものからの多数の変更例を有していてもよいということが理解されるべきである。たとえば、カスタマイズされたハードウェアも使用されてもよく、および／または、特定の要素がハードウェア、（アプレットなどの高移植性ソフトウェアを含む）ソフトウェアまたはそれら両方で実現されてもよい。さらに、ネットワーク入力／出力装置などの他の演算装置への接続が利用されてもよい。

【0051】

コードまたはコードの一部を含むための記憶媒体およびコンピュータ読取可能媒体は、記憶媒体および通信媒体を含む当該技術分野において公知のまたは使用される任意の適切な媒体を含み得て、当該媒体は、コンピュータ読取可能な命令、データ構造、プログラムモジュールまたは他のデータなどの情報の格納のための任意の方法または技術において実現される揮発性および不揮発性の、取外し可能および取外し不可能な媒体などであるが、これらに限定されるものではない。記憶媒体およびコンピュータ読取可能媒体の例としては、所望の情報を格納するために使用可能であり、コンピュータによって読取可能であるRAM、ROM、EEPROM、フラッシュメモリもしくは他のメモリ技術、CD-ROM、デジタル多用途ディスク(digital versatile disk: DVD)もしくは他の光学記憶装置、磁気カセット、磁気テープ、磁気ディスク記憶装置もしくは他の磁気記憶装置、またはその他のメモリ媒体が挙げられる。記憶媒体およびコンピュータ読取可能媒体は、非一時的なメモリ装置を含んでいてもよい。

10

【0052】

本発明の特定の実施形態について説明してきたが、さまざまな変換例、変更例、代替的な構成および等価物も本発明の範囲内に包含される。本発明の実施形態は、特定の具体的なデータ処理環境内での動作に限定されるものではなく、複数のデータ処理環境内で自由に動作できる。さらに、特定の一連のトランザクションおよびステップを使用して本発明の実施形態について説明してきたが、本発明の範囲が、記載されている一連のトランザクションおよびステップに限定されるものではないということが当業者に明らかであるべきである。

20

【0053】

さらに、ハードウェアおよびソフトウェアの特定の組み合わせを使用して本発明の実施形態について説明してきたが、ハードウェアおよびソフトウェアの他の組み合わせも本発明の範囲内であることが認識されるべきである。本発明の実施形態は、ハードウェアのみで実現されてもよく、またはソフトウェアのみで実現されてもよく、またはそれらの組み合わせを使用して実現されてもよい。

30

【0054】

したがって、明細書および図面は、限定的な意味ではなく例示的な意味で考えられるべきである。しかし、特許請求の範囲に記載されているより広範な精神および範囲から逸脱することなく、追加、削減、削除ならびに他の変換および変更がそれに対してなされてもよいということは明白であろう。当該変換および変更は、開示した特徴の任意の関連する組み合わせを含む。

【図1】

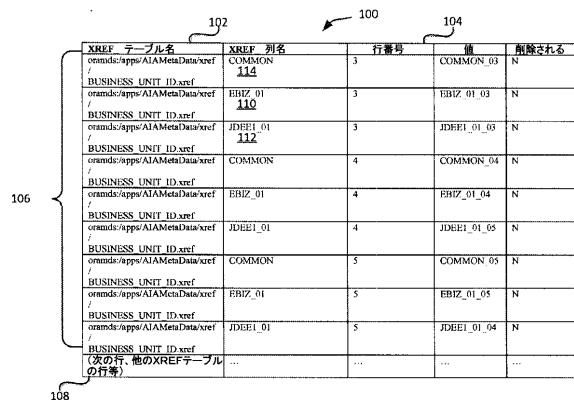


Diagram illustrating the structure of the XREF table. The table has columns: XREF テーブル名, XREF 列名, 行番号, 値, and 削除される. The table contains several rows, with specific rows 102, 104, and 106 highlighted. Row 102 has XREF テーブル名 'oramsd:apps/AIA_MetaData/xref' and XREF 列名 'COMMON_114'. Row 104 has XREF テーブル名 'oramsd:apps/AIA_MetaData/xref' and XREF 列名 'EBIZ_01_110'. Row 106 has XREF テーブル名 'oramsd:apps/AIA_MetaData/xref' and XREF 列名 'JDEE1_01_112'.

| XREF テーブル名 | XREF 列名 | 行番号 | 値 | 削除される |
|-------------------------------|--------------|-----|-------------|-------|
| oramsd:apps/AIA_MetaData/xref | COMMON_114 | 3 | COMMON_03 | N |
| /BUSINESS_UNIT_ID/xref | EBIZ_01_110 | 3 | EBIZ_01_03 | N |
| /BUSINESS_UNIT_ID/xref | JDEE1_01_112 | 3 | JDEE1_01_03 | N |
| /BUSINESS_UNIT_ID/xref | COMMON | 4 | COMMON_04 | N |
| /BUSINESS_UNIT_ID/xref | EBIZ_01 | 4 | EBIZ_01_04 | N |
| /BUSINESS_UNIT_ID/xref | JDEE1_01 | 4 | JDEE1_01_05 | N |
| /BUSINESS_UNIT_ID/xref | COMMON | 5 | COMMON_05 | N |
| /BUSINESS_UNIT_ID/xref | EBIZ_01 | 5 | EBIZ_01_05 | N |
| /BUSINESS_UNIT_ID/xref | JDEE1_01 | 5 | JDEE1_01_04 | N |
| (次の行、他のXREFテーブルの行等) | ... | ... | ... | ... |

FIG. 1

【図2】

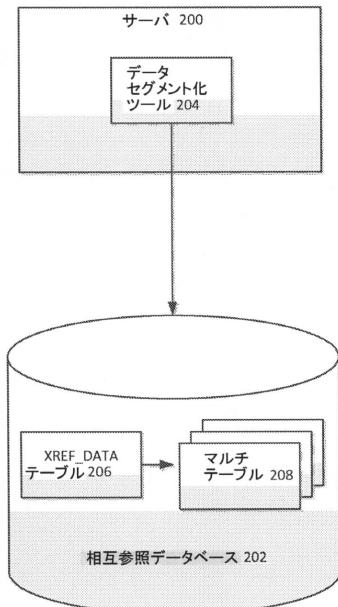


FIG. 2

【図3】

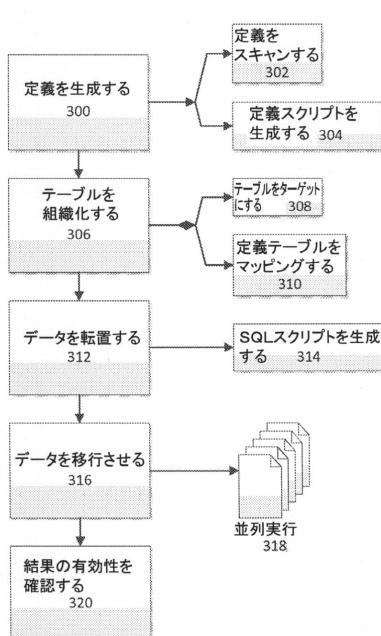


FIG. 3

【図4 A】

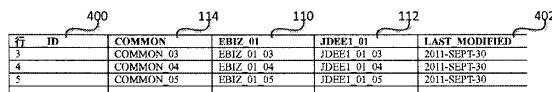


Table A (XREF_DATA table):

| 行 | ID | COMMON | EBIZ_01 | JDEE1_01 | LAST_MODIFIED |
|-----|-----------|------------|-------------|--------------|---------------|
| 3 | COMMON_03 | EBIZ_01_03 | JDEE1_01_03 | 2011-SEPT-30 | |
| 4 | COMMON_04 | EBIZ_01_04 | JDEE1_01_04 | 2011-SEPT-30 | |
| 5 | COMMON_05 | EBIZ_01_05 | JDEE1_01_05 | 2011-SEPT-30 | |
| ... | ... | ... | ... | ... | |

FIG. 4A

【図4 B】

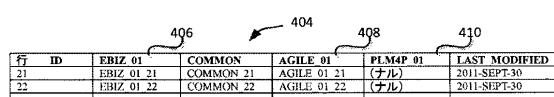


Table B (Mutual table):

| 行 | ID | EBIZ_01 | COMMON | AGILE_01 | PLM4P_01 | LAST_MODIFIED |
|-----|------------|-----------|-------------|----------|--------------|---------------|
| 21 | EBIZ_01_21 | COMMON_21 | AGILE_01_21 | (ナル) | 2011-SEPT-30 | |
| 22 | EBIZ_01_22 | COMMON_22 | AGILE_01_22 | (ナル) | 2011-SEPT-30 | |
| ... | ... | ... | ... | ... | ... | |

FIG. 4B

【図4 C】

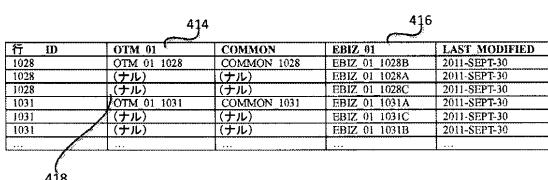


Table C (XREF_DATA table):

| 行 | ID | OTM_01 | COMMON | EBIZ_01 | LAST_MODIFIED |
|------|-------------|-------------|---------------|--------------|---------------|
| 1028 | OTM_01_1028 | COMMON_1028 | EBIZ_01_1028B | 2011-SEPT-30 | |
| 1028 | (ナル) | (ナル) | EBIZ_01_1038A | 2011-SEPT-30 | |
| 1028 | (ナル) | (ナル) | EBIZ_01_1028C | 2011-SEPT-30 | |
| 1031 | OTM_01_1031 | COMMON_1031 | EBIZ_01_1031A | 2011-SEPT-30 | |
| 1031 | (ナル) | (ナル) | EBIZ_01_1031C | 2011-SEPT-30 | |
| 1031 | (ナル) | (ナル) | EBIZ_01_1031B | 2011-SEPT-30 | |
| ... | ... | ... | ... | ... | |

FIG. 4C

【図5】

| 列名 | 目的 |
|---------------------------|--|
| XREF_TABLE_NAME | 502 XREFアーティファクト名 |
| TOTAL_NUM_REGULAR_ROWS | 504 1対多のケースに該当しないこのアーティファクトについてのXREF_DATA内の行の数 |
| NUM_MIGRATED_REGULAR_ROWS | 506 対応するマルチテーブルにすでに移行された1対多の行の数 |
| TOTAL_NUM_MULTI_ROWS | 508 1対多のケースに該当するこのアーティファクトについてのXREF_DATA内の行の数 |
| NUM_MIGRATED_MULTI_ROWS | 510 対応するマルチテーブルにすでに移行された1対多の行の数 |
| STARTED_PROCESSING | 512 このアーティファクトについての移行が開始した時間 |
| FETCH_ANOMALIES_START | 514 1対多の行のクエリが開始した時間 |
| FETCH_ANOMALIES | 516 1対多の行のクエリに要した時間 |
| BULK_MIGRATE_START | 518 「正規の」行の移行が開始した時間 |
| BULK_MIGRATE | 520 「正規の」行を移行させるのに要した時間 |
| PROCESS_ANOMALIES_START | 522 1対多の行の移行が開始した時間 |
| PROCESS_ANOMALIES | 524 1対多の行を移行させるのに要した時間 |
| HAD_EXCEPTION | 526 このアーティファクトの移行の処理中に例外が起これば非ナル |

FIG. 5

【図6】

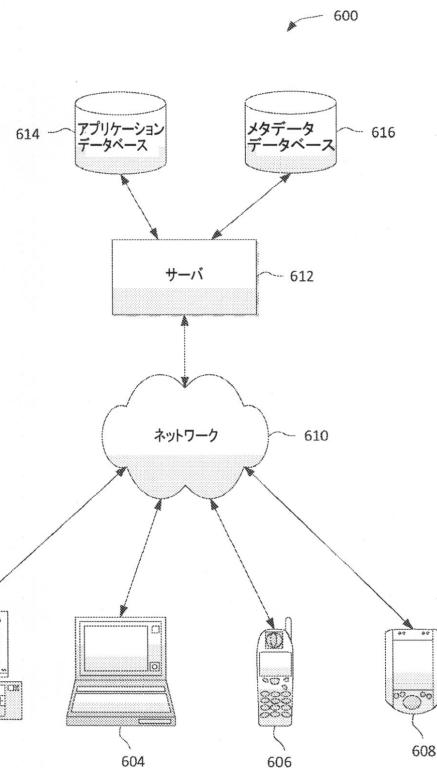


FIG. 6

【図7】

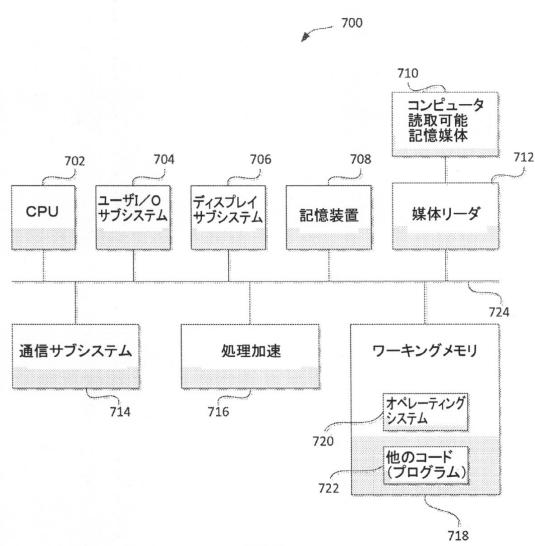


FIG. 7

フロントページの続き

(72)発明者 ン , アーノルド
アメリカ合衆国、95014 カリフォルニア州、クバチーノ、ブレット・アベニュー、10095

(72)発明者 モディ , ラジヤン
アメリカ合衆国、80130 コロラド州、ハイランズ・ランチ、グレンゲート・ループ、108
44

(72)発明者 ガウ , ジョニー
アメリカ合衆国、94539 カリフォルニア州、フリーモント、アマリロ・コート、48725

審査官 甲斐 哲雄

(56)参考文献 特開平11-110262 (JP, A)
特開2008-171225 (JP, A)
特開2006-293981 (JP, A)

(58)調査した分野(Int.Cl. , DB名)

G 06 F 12 / 00