(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6258549号
(P6258549)
(45)発行日　平成30年1月10日(2018.1.10)　(24)登録日　平成29年12月15日(2017.12.15)

(51)Int.Cl.　F I
G06F 17/30 (2006.01)　G06F 17/30 220Z
G06F 17/30 110C

請求項の数 31　(全 26 頁)

(21)出願番号　特願2017-153599(P2017-153599)
(22)出願日　平成29年8月8日(2017.8.8)
(62)分割の表示　特願2016-502660(P2016-502660)の分割
原出願日　平成26年3月14日(2014.3.14)
(65)公開番号　特開2017-224331(P2017-224331A)
(43)公開日　平成29年12月21日(2017.12.21)
審査請求日　平成29年8月30日(2017.8.30)
(31)優先権主張番号　13/835,590
(32)優先日　平成25年3月15日(2013.3.15)
(33)優先権主張国　米国(US)

早期審査対象出願

(73)特許権者　515160389
インフォマティカ　エルエルシー
アメリカ合衆国　カリフォルニア州　９４０６３、レッドウッド　シティ、シーポート　ブルバード　２１００
２１００　Ｓｅａｐｏｒｔ　Ｂｌｖｄ，Ｒｅｄｗｏｏｄ　Ｃｉｔｙ，ＣＡ　９４０６３　Ｕ．Ｓ．Ａ.
(74)代理人　100120662
弁理士　川上　桂子
(72)発明者　グロンディン、　リチャード
カナダ　ジェイ３イー　２ゼット１、ケベック、サンテ－ジュリ、リュ　ヴォクラン　３

最終頁に続く

(54)【発明の名称】個別のデータ値に対する効率よい演算を行うための方法、装置、並びにコンピュータ可読媒体

(57)【特許請求の範囲】
【請求項１】
一または複数の演算装置によって個別のデータ値に対して演算を効率良く行うための方法であって、前記方法は、
前記一または複数の演算装置の少なくとも一つが、テーブル内のデータカラムを宛先とした照会を受領する工程であって、前記データカラムはデータベースのドメインに対応し、前記照会は、前記照会に応答して検索されたデータをグループ化するための一または複数のグループセットを定義し、前記一または複数のグループセットのそれぞれのグループは、前記データベースの一または複数の他のドメインに関連付けられた一または複数の値の固有のグループに対応している、工程と、
前記一または複数の演算装置の少なくとも一つが、前記テーブルにおいて前記グループセットに対応する一または複数の値の固有のグループを含む列に存在する前記データカラム内のいずれかの値を特定することにより、前記一または複数のグループセット中の各グループセットについてエンティティマップベクトルを生成する工程であって、各エンティティマップベクトルの長さは前記データカラムに対応する前記ドメイン内の固有データ値の数と等しく、各エンティティマップベクトル内の各ビット位置は、固有データ値が語彙順に配列された一覧中の対応する固有データ値の語彙順位置に対応し、各エンティティマップベクトル中の各ビットの値は、前記グループセットにおける対応する固有データ値の有無を示す工程と、
を含む、方法。

【請求項２】
エンティティマップベクトル中のビットが「１」という値であれば前記グループセットに固有データ値が有ることを示し、エンティティマップベクトル中のビットが「０」という値であれば前記グループセットに固有データ値が無いことを示す、請求項１に記載の方法。
【請求項３】
グループセット中の個別のデータ値の数は、当該グループセットに対応するエンティティマップベクトルの前記ビットの各々の値の合計に等しい、請求項１に記載の方法。
【請求項４】
前記一または複数のグループセットは少なくとも二つのグループセットを含み、前記方法は、
前記グループセットの二つ以上からのデータを集計して、合成グループセットにするためのコマンドを、前記一または複数の演算装置の少なくとも一つが受領する工程と、
前記二つ以上のグループセットに対応するエンティティマップベクトルについて、前記一または複数の演算装置の少なくとも一つが、ブールＯＲ演算を行い、合成エンティティマップベクトルを生成する工程と、
をさらに含む、請求項１に記載の方法。
【請求項５】
前記合成グループセット中の個別のデータ値の数は、前記合成エンティティマップベクトルの前記ビットの各々の値の合計に等しい、請求項４に記載の方法。
【請求項６】
前記テーブルはパーティション化され複数の演算ノードにわたって分散されており、前記データカラムはカラムパーティションのセットの中の第一カラムパーティションであり、前記ドメインはドメインパーティションのセットの中の第一ドメインパーティションであり、前記一または複数のエンティティマップベクトルは一または複数の第一パーティションエンティティマップベクトルであり、
前記方法は、
前記一または複数の演算装置の少なくとも一つが、前記第一ドメインパーティションを前記ドメインパーティションのセットの中の一または複数のその他のドメインパーティションと合成し、グローバルドメインを生成する工程と、
前記一または複数の演算装置の少なくとも一つが、前記一または複数の第一パーティションエンティティマップベクトルに対応する一または複数のグローバルエンティティマップベクトルを生成する工程であって、各グローバルエンティティマップベクトルの長さは前記グローバルドメインにおける固有データ値の数に等しく、前記グローバルエンティティマップベクトルにおける各ビット位置は、前記グローバルドメイン内の固有データ値が語彙順に配列された一覧中の対応する固有データ値の語彙順位置に対応し、前記エンティティマップベクトル内の各ビットの値は、前記グループセットにおける対応する固有データ値の有無を示す、工程と、
をさらに含む、請求項１に記載の方法。
【請求項７】
前記一または複数の演算装置の少なくとも一つが、前記第一パーティションに対応する前記一または複数のグローバルエンティティマップベクトルと、第二パーティションに対応する一または複数の第二グローバルエンティティマップベクトルとに、ブールＯＲ演算を行い、一または複数の合成グローバルエンティティマップベクトルを生成する工程をさらに含む、請求項６に記載の方法。
【請求項８】
前記合成グローバルエンティティマップベクトルの各々の各ビットの値の合計は、前記第一パーティションと前記第二パーティションとにわたる当該グローバルエンティティマップベクトルに対応するあるグループセットの中の個別の要素の総数である、請求項７に記載の方法。

【請求項９】
グループセットの数は、少なくとも部分的に、前記テーブル中の第二データカラム中の個別のデータ値の数によって決定される。請求項１に記載の方法。
【請求項１０】
前記テーブル中の一または複数の第二データカラムを参照して前記グループセットが定義される、請求項１に記載の方法。
【請求項１１】
個別のデータ値に対して演算を効率良く行うための装置であって、前記装置は、
一または複数のプロセッサと、
前記一または複数のプロセッサの少なくとも一つと動作的に結合され、命令を保存する、一または複数のメモリとを備え、前記命令は、前記一または複数のプロセッサの少なくとも一つによって実行される時に、前記一または複数のプロセッサの少なくとも一つに、
テーブル内のデータカラムを宛先とした照会を受領し、前記データカラムはデータベースのドメインに対応し、前記照会は、前記照会に応答して検索されたデータをグループ化するための一または複数のグループセットを定義し、前記一または複数のグループセットのそれぞれのグループは、前記データベースの一または複数の他のドメインに関連付けられた一または複数の値の固有のグループに対応し、
前記テーブルにおいて前記グループセットに対応する一または複数の値の固有のグループを含む列に存在するデータカラム内のいずれかの値を特定することにより、前記一または複数のグループセット中の各グループセットについてエンティティマップベクトルを生成し、各エンティティマップベクトルの長さは前記データカラムに対応する前記ドメイン内の固有データ値の数と等しく、各エンティティマップベクトル中の各ビットの位置は、前記固有データ値が語彙順に配列された一覧中の対応する固有データ値の語彙順位置に対応し、各エンティティマップベクトル中の各ビットの値は、前記グループセットにおける対応する固有データ値の有無を示す、
ようにさせる装置。
【請求項１２】
エンティティマップベクトル中のビットが「１」という値であれば前記グループセットに固有データ値が有ることを示し、エンティティマップベクトル中のビットが「０」という値であれば前記グループセットに固有データ値が無いことを示す、請求項１１に記載の装置。
【請求項１３】
グループセット中の個別のデータ値の数は、当該グループセットに対応する前記エンティティマップベクトルの前記ビットの各々の値の合計に等しい、請求項１１に記載の装置。
【請求項１４】
前記一または複数のグループセットは複数のグループセットを含み、前記一または複数のメモリはさらに命令を保存し、前記命令は、前記一または複数のプロセッサの少なくとも一つによって実行される時に、前記一または複数のプロセッサの少なくとも一つに、
前記グループセットの二つ以上からのデータを集計して、合成グループセットにするためのコマンドを受領し、
前記二つ以上のグループセットに対応するエンティティマップベクトルについて、ブールＯＲ演算を行い、合成エンティティマップベクトルを生成する、
ようにさせる、請求項１１に記載の装置。
【請求項１５】
前記合成グループセット中の個別のデータ値の数は、前記合成エンティティマップベクトルの前記ビットの各々の値の合計に等しい、請求項１４に記載の装置。
【請求項１６】
前記テーブルはパーティション化され複数の演算ノードにわたって分散されており、前記データカラムはカラムパーティションのセットの中の第一カラムパーティションであり

、前記ドメインはドメインパーティションのセットの中の第一ドメインパーティションであり、前記一または複数のエンティティマップベクトルは一または複数の第一パーティションエンティティマップベクトルであって、前記一または複数のメモリはさらに命令を保存し、前記命令は、前記一または複数のプロセッサの少なくとも一つによって実行される時に、前記一または複数のプロセッサの少なくとも一つに、

前記第一ドメインパーティションを前記ドメインパーティションのセットの中の一または複数のその他のドメインパーティションと合成し、グローバルドメインを生成し、

前記一または複数の第一パーティションエンティティマップベクトルに対応する一または複数のグローバルエンティティマップベクトルを生成し、

各グローバルエンティティマップベクトルの長さは前記グローバルドメイン内の固有データ値の数に等しく、前記グローバルエンティティマップベクトル中の各ビットの位置は、前記グローバルドメイン中の前記固有データ値が語彙順に配列された一覧中の対応する固有データ値の語彙順位置に対応し、前記エンティティマップベクトルの各ビットの値は、前記グループセット中における対応する固有データ値の有無を示す、

ようにさせる、請求項１１に記載の装置。

【請求項１７】

前記一または複数のメモリはさらに命令を保存し、前記命令は、前記一または複数のプロセッサの少なくとも一つによって実行される時に、前記一または複数のプロセッサの少なくとも一つに、

前記第一パーティションに対応する前記一または複数のグローバルエンティティマップベクトルと、第二パーティションに対応する一または複数の第二グローバルエンティティマップベクトルとに、ブールＯＲ演算を行い、一または複数の合成グローバルエンティティマップベクトルを生成するようにさせる、

請求項１６に記載の装置。

【請求項１８】

前記合成グローバルエンティティマップベクトルの各々の各ビットの値の合計は、前記第一パーティションと前記第二パーティションとにわたる当該グローバルエンティティマップベクトルに対応するあるグループセットの中の個別の要素の総数である、請求項１７に記載の装置。

【請求項１９】

グループセットの数は、少なくとも部分的に、前記テーブル中の第二データカラム中の個別のデータ値の数によって決定される、請求項１１に記載の装置。

【請求項２０】

前記テーブル中の一または複数の第二データカラムを参照して前記グループセットが定義される、請求項１１に記載の装置。

【請求項２１】

コンピュータ可読命令を含むコンピュータプログラムであって、前記命令は、一または複数の演算装置によって実行される時に前記一または複数の演算装置の少なくとも一つに、

テーブル内のデータカラムを宛先とした照会を受領し、前記データカラムはデータベースのドメインに対応し、前記照会は、前記照会に応答して検索されたデータをグループ化するための一または複数のグループセットを定義し、前記一または複数のグループセットのそれぞれのグループは、前記データベースの一または複数の他のドメインに関連付けられた一または複数の値の固有のグループに対応し、

前記テーブルにおいて前記グループセットに対応する一または複数の値の固有のグループを含む列に存在する前記データカラム内のいずれかの値を特定することにより、前記一または複数のグループセット中の各グループセットについてエンティティマップベクトルを生成し、各エンティティマップベクトルの長さは前記データカラムに対応する前記ドメイン内の固有データ値の数と等しく、各エンティティマップベクトル内の各ビット位置は、固有データ値が語彙順に配列された一覧中の対応する固有データ値の語彙順位置に対応

し、各エンティティマップベクトル中の各ビットの値は、前記グループセットにおける対応する固有データ値の有無を示す、
ようにさせる、コンピュータプログラム。
【請求項２２】
エンティティマップベクトル中のビットが「１」という値であれば前記グループセットに固有データ値が有ることを示し、エンティティマップベクトル中のビットが「０」という値であれば前記グループセットに固有データ値が無いことを示す、請求項２１に記載のコンピュータプログラム。
【請求項２３】
グループセット中の個別のデータ値の数は、当該グループセットに対応するビットマップベクトルの前記ビットの各々の値の合計に等しい、請求項２１に記載のコンピュータプログラム。
【請求項２４】
前記一または複数のグループセットは複数のグループセットを含み、前記コンピュータプログラムはさらに追加の命令を含み、前記追加の命令は、前記一または複数の演算装置によって実行される時に、前記一または複数の演算装置の少なくとも一つに、
前記グループセットの二つ以上からのデータを集計して、合成グループセットにするためのコマンドを受領し、
前記二つ以上のグループセットに対応するエンティティマップベクトルについて、ブールＯＲ演算を行い、合成エンティティマップベクトルを生成する、
ようにさせる、請求項２１に記載のコンピュータプログラム。
【請求項２５】
前記合成グループセット中の個別のデータ値の数は、前記合成エンティティマップベクトルの前記ビットの各々の値の合計に等しい、請求項２４に記載のコンピュータプログラム。
【請求項２６】
前記テーブルはパーティション化され複数の演算ノードにわたって分散されており、前記データカラムはカラムパーティションのセットの中の第一カラムパーティションであり、前記ドメインはドメインパーティションのセットの中の第一ドメインパーティションであり、前記一または複数のエンティティマップベクトルは一または複数の第一パーティションエンティティマップベクトルであって、前記コンピュータプログラムはさらに追加の命令を含み、前記追加の命令は、一または複数の演算装置によって実行される時に前記一または複数の演算装置の少なくとも一つに、
前記第一ドメインパーティションを前記ドメインパーティションのセットの中の一または複数のその他のドメインパーティションと合成し、グローバルドメインを生成し、
前記一または複数の第一パーティションエンティティマップベクトルに対応する一または複数のグローバルエンティティマップベクトルを生成し、
各グローバルエンティティマップベクトルの長さは前記グローバルドメイン内の固有データ値の数に等しく、前記グローバルエンティティマップベクトル中の各ビットの位置は、前記グローバルドメイン中の前記固有データ値が語彙順に配列された一覧中の対応する固有データ値の語彙順位置に対応し、前記エンティティマップベクトルの各ビットの値は、前記グループセット中における対応する固有データ値の有無を示す、
ようにさせる、請求項２１に記載のコンピュータプログラム。
【請求項２７】
前記コンピュータプログラムはさらに追加の命令を含み、前記追加の命令は、一または複数の演算装置によって実行される時に前記一または複数の演算装置の少なくとも一つに、
前記第一パーティションに対応する前記一または複数のグローバルエンティティマップベクトルと、第二パーティションに対応する一または複数の第二グローバルエンティティマップベクトルとに、ブールＯＲ演算を行い、一または複数の合成グローバルエンティテ

ィマップベクトルを生成させる、請求項２６に記載のコンピュータプログラム。
【請求項２８】
前記合成グローバルエンティティマップベクトルの各々の各ビットの値の合計は、前記第一パーティションと前記第二パーティションとにわたる当該グローバルエンティティマップベクトルに対応するあるグループセットの中の個別の要素の総数である、請求項２７に記載のコンピュータプログラム。
【請求項２９】
グループセットの数は、少なくとも部分的に、前記テーブル中の第二データカラム中の個別のデータ値の数によって決定される、請求項２１に記載のコンピュータプログラム。
【請求項３０】
前記テーブル中の一または複数の第二データカラムを参照して前記グループセットが定義される、請求項２１に記載のコンピュータプログラム。
【請求項３１】
請求項２１～３０のいずれか一項に記載のコンピュータプログラムを記録した、コンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】
【関連出願データ】
【０００１】
本願は、２０１３年3月１５日出願の米国特許出願第１３／８３５，５９０号に対して優先権を主張するものであり、この参照によりその全内容が本明細書に組み込まれるものである。
【背景技術】
【０００２】
企業は頻繁にデータをデータウェアハウスに保存する。かかるデータウェアハウスは大量のデータを多数のデータベースおよびデータベーステーブルに分散させて有することが多い。したがって、このような多数のテーブルやデータベースからデータをまとめるためには、データ集計技術を利用しなければならない。
【０００３】
データウェアハウスの集計プロセスの性能を改善するために、下位のデータのサマリを保存する下位集計（Low Level Aggregate、LLA）テーブルがよく使用される。多数のパーティションに保存される分散データベースの場合、各パーティションにＬＬＡテーブルが利用でき、集計プロセスは各ＬＬＡテーブルからのデータをくみあわせてデータベース全体についての集計されたテーブルを生成することができる。このような集計プロセスは、ロールアッププロセスとして知られ、ソースデータからではなく下位のサマリ（ＬＬＡテーブル）から上位のサマリが生成できるので有用である。
【０００４】
ロールアッププロセスは、超並列処理（ＭＰＰ）データベースアーキテクチャにおいて多数のデータパーティションに対して集計クエリを実行するために、つまりパーティション化されたテーブルについて照会を行うために利用できる。
【０００５】
ロールアッププロセス中に多数のテーブルからデータを正確に集計するためには、集計が行われている根底にある関数が加法でなければならない。例えば、第一のテーブルおよび第二のテーブル両方ともに出費額に対応するカラムがあり、ユーザが両方のテーブルの出費の総額の合計を計算しようとすると、第一テーブルの出費額の合計と、第二テーブルの出費額の合計とを加算で集計し、正確な総額を出すことができる。しかしながら、ロールアッププロセスでユーザが集計しようとする関数すべてが、加法関数というわけではない。
【図面の簡単な説明】
【０００６】

【図１】図１Ａ～１Ｇは、客の来店記録が入っている見本データベーステーブルと、見本データベーステーブルから生成される下位集計テーブルの例を示す。
【図２】図２Ａ～２Ｂは、開示された実施形態にかかる、個別の（distinct）データ値について効率良く演算を行う方法のフローチャートである。
【図３】図３Ａ～３Ｃは、開示された実施形態にかかる、テーブル中のデータの一または複数のカラムをトークン化する工程を示すテーブルである。
【図４】図４Ａ～４Ｃは、トークン化されたテーブルから生成されたＬＬＡテーブルと、ＬＬＡテーブルから生成された二つの見本上位集計テーブルとを示す。
【図５Ａ】図５Ａは、開示された実施形態にかかる、エンティティマップベクトルデータ構造を利用するトークン化されたテーブルから生成されたＬＬＡテーブルを示す。
【図５Ｂ】図５Ｂは、開示された実施形態にかかる、エンティティマップベクトル中のビットの各々に値を割り当てる方法を示す。
【図５Ｃ】図５Ｃは、開示された実施形態にかかる、ドメインの語彙順配列の固有データ値のテーブルを示す。
【図５Ｄ】図５Ｄは、開示された実施形態にかかる、エンティティマップベクトル中のビットの各々に値を割り当てる方法を示す。
【図６Ａ】図６Ａは、開示された実施形態にかかる、図５ＡのＬＬＡテーブルに保存されたエンティティマップベクトル上でロールアッププロセスを行う時の工程を示す。
【図６Ｂ】図６Ｂは、図６Ａに示されるロールアッププロセスの結果である上位テーブルを示す。
【図７】図７Ａ～７Ｂは、開示された実施形態にかかる、多数のパーティションにまたがってドメインを結合することにより分散データベース中の個別のデータ値について演算を効率良く行うためのフローチャートを示す。
【図８】図８Ａ～８Ｄは、開示された実施形態にかかる、パーティションに特定されたエンティティＩＤの見本セットをグローバルエンティティＩＤへ変換するために必要な工程のそれぞれを示す。
【図９】図９は、開示された実施形態にかかる、多数のパーティションにまたがってグローバルエンティティマップベクトルを保存するためのデータキューブを示す。
【図１０】図１０は、開示された実施形態にかかる、経時的に割り当てられたエンティティＩＤを使用したデータベースウェアハウス更新演算を示す。
【図１１Ａ】図１１Ａは、開示された実施形態にかかる、ドメイン同期データ構造と、プロセスフローとを示す。
【図１１Ｂ】図１１Ｂは、マッピングテーブルを使用してローカルエンティティマップベクトルをグローバルエンティティマップベクトルにマッピングする時のマッピングを示す。
【図１２】図１２は、開示された実施形態にかかる、個別のデータ値について演算を効率良く行うための方法を実施するために使用できるコンピューティング環境の一例を示す。
【発明を実施するための形態】
【０００７】
方法、装置、およびコンピュータ可読媒体が実施例および実施形態として本明細書に記載されているが、個別のデータ値に対する効率よい演算を行うための方法、装置、およびコンピュータ可読媒体が、実施形態または図面に記載されたものに限定されないことを、当業者は認識する。図面および明細書が開示された特定の形態に限定されることを意図したものではないことを理解されたい。むしろ、添付の請求項の精神及び範囲から逸脱しないすべての変形、均等物、および代替物を包含することを意図している。本明細書中に使用された見出しはいずれも、整理するという目的のみのために使用されており、本明細書または請求項の範囲を制限することを企図していない。本明細書で使用されている通り、「～することができる（can）」という表現は、必須の意味（すなわち、「～でなければならない（must）」という意味）ではなく、許容の意味（すなわち、可能性があるという意味）で用いられている。同様に、「含む」（「include」「including」、および「incl

udes」）という用語は、「含む」ことを意味しており、それに限定されるという意味ではない。

【０００８】

図１Ａのテーブルを参照し、加法関数用のロールアッププロセスのいくつかの例を以下に説明する。図１Ａは、客の来店記録が入っている見本データベーステーブルを示す。このテーブルは、客の名前と、店舗ＩＤと、来店中の出費額とのカラムを有する。

【０００９】

本明細書全体で用いられる照会言語は、構造化照会言語（ＳＱＬ）であるが、いかなるデータベース照会言語も、本発明の範囲から逸脱することなく使用でき、その例としては、コンテクスチュアル照会言語（CQL）、XQuery、YQL、Datalog、OQL、RDQL、多次元式（MDX）など多数の言語が挙げられることが理解される。

【００１０】

図１Ｂには、各店舗についての来店数のＬＬＡテーブルが示されている。「来店数」カラムは、店舗のそれぞれについての記録数をカウントすることにより生成され、店舗１には来店数３、店舗２には来店数４という結果になっている。全店舗の総来店数を特定するためにこのテーブルを使用してロールアッププロセスを実行しようとすると、関数Sum(来店数)でデータがロールアップでき、総数７という結果になる。この場合、来店の総数を特定するために元のテーブルへ戻る必要が無い。

【００１１】

同様に、図１Ｃのテーブルは、店舗それぞれの総出費額のＬＬＡテーブルである。この出費額は、各店舗用の元のテーブルの「出費額」カラムの値を合計することにより特定される。ユーザが全店舗の総出費額を特定するためにこのテーブルをロールアップしようとすれば、図１ＣのＬＬＡテーブル上の関数Sum(出費額)を使用して、１０７という正確な総額に至ることができる。

【００１２】

ロールアッププロセス中のＬＬＡテーブルの集計は、単純な合計以外の関数についても行うことができる。図１Ｄは、各店舗における来店ごとの平均出費額のＬＬＡテーブルを示す。この額は、各店舗についての元のテーブルの「出費額」カラムの値を平均することによって特定される。さらに、ＬＬＡテーブルには、前述のように各店舗への来店数のカウントが入っている。ユーザが、全店舗での平均出費額を特定するためにこのテーブルをロールアップしたいと思えば、下記の関数を使用することができる。

【００１３】

【数１】

$$平均出費額(合計)=\frac{\left((平均_{店舗1}\times 来店数_{店舗1})+(平均_{店舗2}\times 来店数_{店舗2})\right)}{Sum\ (来店数)}$$

【００１４】

繰り返しになるが、ロールアッププロセスは、元のテーブルを必要とせず、かつ、ＬＬＡテーブルのみを使用することによって行える。

【００１５】

図１Ｅおよび１Ｆは、それぞれ、各店舗における最小出費額が入っているＬＬＡテーブルと、各店舗における最大出費額とが入っているＬＬＡテーブルとを示す。いずれかの店舗での最小の出費額を特定するために図１Ｅのテーブルをロールアップするためには、ユーザはＬＬＡテーブルについて関数Min(最小出費額)を使用でき、正確な答えである５に至ることができる。同様に、いずれかの店舗での最大の出費額を特定するために図１Ｆのテーブルをロールアップするためには，ユーザはＬＬＡテーブルについて関数Max(最大出費額）を使用でき、正確な答えである３１に至ることができる。

【００１６】
図１Ｇを参照すると、各店舗を訪れたことのある個別の（distinct）客の数のカウントが入っているＬＬＡテーブルが示されている。これは、店舗ＩＤでグループ分けされたCount(distinct<客の名前>)関数を使用して行うことができる。したがって、この場合、３人の個別の客（Bill、George、および Larry） が店舗１に来店し、４人の個別の客（Bill、George、Max、および Larry）が店舗２に来店していた。
【００１７】
ユーザが図１ＧのＬＬＡテーブルをロールアップしたいなら、根底にある主なテーブルである図１Ａのテーブルにアクセスしないで店舗１または店舗２によく来店する個別の客の総数を正しく特定する方法は無い。ＬＬＡテーブルの個別の客のカラム上で機能するcount-distinct関数は、２つの個別の値の総数を返信するだけであり、合計として７を返すだろうが、両方の店舗を訪れた個別の客の正確な数は、４人である。同様に、もしＬＬＡテーブルに各々の店舗で出費された個別の出費ドル額の合計が入っており、ユーザが、全ての店舗で出費された個別のドル額のすべての合計を計算したければ、元のテーブルにアクセスしなければならないだろう。さもなければ、店舗１で出費された１１ドルと店舗２で出費された１１ドルが二重に計上されてしまうだろうからである。さらに、もしＬＬＡテーブルに各店舗で出費された個別の出費額すべてについての平均が入っていれば、両店舗での個別の出費額すべてについての平均をロールアップで特定しようとしてもうまくいかず、根底にある主なテーブルにアクセスしなければならないだろう。
【００１８】
count<distinct>のような非加法コマンドを含んでいる照会は、ＬＬＡテーブルのロールアップ演算を使用することが出来ず、その結果、計算コストが非常に高くなる。なぜなら、これらコマンドを、根底にあるデータについて実行する必要があるからである。パーティション化データベースか、または分散データベースにおいては、正確な値を検索するためには根底にあるデータが集計されているかまたはパーティション間で交換されなければならないので、このコストは膨れ上がる。例えば、もしカラムＸＹＺが二つのパーティションに散らばっており、ユーザがカラムＸＹＺの個別の値の数を特定したいなら、第一パーティション上のカラムＸＹＺの値のすべてを、第二パーティションのカラムＸＹＺの値のすべてと共に、何らかの方法で一緒に管理して、個別の値の数を特定しなければならない。
【００１９】
出願人は、エンティティマップベクトル（ＥＭＶ）と称される方法と、関連したデータ構造とを発見した。これは、ユーザが非加法集計関数を加法として使用できるようにし、それによって、以前はＬＬＡテーブルよりも下位であるテーブルにある、根底にあるデータへのアクセスや処理を必要とした関数について、ＬＬＡテーブル上でロールアッププロセスが実行可能になった。出願人の技術によってさらに、パーティション化されたデータベースのノード間で交換されるデータの量が大幅に削減できるようになった。これは、各ノードについてのＬＬＡテーブルが、非加法集計関数についても、加法集計できるからである。
【００２０】
図２Ａは、開示された実施形態にかかる、ＥＭＶを生成する方法のフローチャートを示す。ステップ２０１で、データベースのテーブル中のデータのトークン化されたカラムが、ＥＭＶを生成する前に保存される。このトークン化プロセスを、図３Ａ－３Ｃを参照して説明する。さらに、図２Ｂに示され、以下に記載されるように、ＥＭＶは、データのトークン化されたカラムを使用することなく、あるドメインでの固有データ値の語彙順に基づいて生成することができる。
【００２１】
図３Ａは、トークン化前の顧客取引テーブル３００を示す。このテーブルは、タプルＩＤ（TupleID）３０１と、週（Week）３０２、店舗ＩＤ（SstoreID）３０３、客ＩＤ（CustID）３０４、製品ＩＤ（productID）３０５、量（Qty）３０６、および出費額（Spent）

３０７のデータカラムとを備える。データカラムの各々は、関連付けられたドメインに対応させてもよい。例えば、客ＩＤを備えたドメインは、図示されたテーブルの客ＩＤカラムに入力するために使用できるが、同じドメインを、客のアドレスと共に客ＩＤを一覧にした別のテーブルに入力するために使用できる。もちろん、ドメインは、同じテーブルの複数のカラムに入力するためにも使用できる。例えば、都市名のドメインは、起点都市と目的地都市とに対応するテーブルのカラムに入力するために使用できる。したがって、週３０２のカラムはドメインD_Weekに対応でき、店舗ＩＤ３０３はドメインD_StoreIdに対応でき、客ＩＤ３０４はドメインD_CustIdに対応でき、製品ＩＤ３０５はドメインD_ProductIdに対応でき、量３０６はドメインD_Qtyに対応でき、出費額３０７はドメインD_Spentに対応できる。

【００２２】

各データカラムのデータ値をトークン化するために、各ドメインに生じる固有データ値（異なるカラムに対応する）は、識別されて固有のトークン値にマッピングされなければならない。固有データ値をテーブル中の各ドメインの固有のトークン値にマッピングするが、図３Ｂのマッピングテーブル３１０はこのマッピングを示している。各ドメインの各固有データ値は、エンティティＩＤ、すなわちＥＩＤと称される、そのドメイン用の固有のトークンにマッピングされる。したがって、例えば、D_Week３１２は３つの固有データ値を備え、したがってこれら３つの固有データ値が３つの対応する固有のＥＩＤにマッピングされる。同様に、D_StoreID３１３もまた３つの固有データ値と３つのＥＩＤとを備え、D_CustId３１４は４つの固有データ値と４つのＥＩＤとを備え、D_ProductIdは４つの固有データ値と４つのＥＩＤとを備え、D_Qtyは５つの固有データ値と５つのＥＩＤとを備え、D_Spentは６個の固有データ値と６個のＥＩＤとを備える。

【００２３】

ドメイン中の固有データ値を対応するＥＩＤにマッピングするために用いられるマッピングテーブルは、ルックアップ辞書として保存できるので、関数および関連付けられた処理は、必要であればＥＩＤを使用してデータ値を検索できる。さらに、ＥＩＤに対する固有データ値のマッピングは、固有データ値の語彙順で生じるものとして示されており、これは割り当てプロセスや、その後のデータの保存や検索を簡便にするからであるが、ＥＩＤの固有データ値への割り当ては固有データ値のいかなる順序で生じてもよい。例えば、第一データ値が第二のデータ値より以前に遭遇すれば下位のＥＩＤを有するよう、経時的な順序でデータ値に割り当てられるようにＥＩＤを割り当ててもよい。

【００２４】

図３Ｃは、トークン化後の顧客取引テーブル３２０を示す。マッピングテーブル３１０からのＥＩＤは、データのカラムの各々においてデータ値に置き換えられている。しかしながら、この高度な暗号化構造においてすら、テーブル３２０を介して生成されたＬＬＡテーブルには、非加法の（distinctタイプの）集計関数を含むロールアッププロセスを実行するのに適切なフォーマットでデータが保存されることはない。

【００２５】

例えば、ＬＬＡテーブルは、週と、店舗ＩＤと、各週に各店舗に来店した個別の客とを一覧にした取引テーブル（「Trx」）３２０に基づいて、下記の定義を用いて、生成できる。

【００２６】

```
Create Table LLA_TRX as
Select      week,
      StoreId,
      Count(distinct CustId) as CD_CustId,
From Trx
Group By week, StoreId;
```

この定義は、各週に各店舗に来店した個別の客に対応する値を、カラムCD_CustIdに割り当てる。さらに、「group-by」コマンドは、週-店舗ＩＤのペアに対応するグループセ

ットを作る。例えば、グループセット１（週１、店舗１）、グループセット２（週１、店舗２）などである。グループセットの構成や数はどんなものでもよい。例えば、グループセットは、一つのカラム、または二つ以上のカラムに基づいていてもよい。

【００２７】

図４Ａには、結果として得られたＬＬＡ＿ＴＲＸテーブル４００が示されている。このテーブルは、グループセットごとの個別の客の数を正確に表している。例えば、テーブルは、週１に店舗１を訪れた個別の客の数を２であると、または、週２に店舗１を訪れた個別の客の数を３であると、正確に一覧にしている。

【００２８】

もしユーザが各週についてテーブルＬＬＡ＿ＴＲＸをロールアップして各週に店舗のいずれかを訪れた個別の客の数を一覧にしようとしたなら、各グループセットについて個別の客の数の正確なカウントを作ることはできないであろう。この場合のグループセットとは、グループ１（週１）、グループ２（週２）、などである。例えば、上位ＬＬＡテーブルを作り出すために下記の定義が用いられたとする。

【００２９】

```
Create Table LLA_TRX_WEEK as
Select      Week,
      Count(distinct CD_CustId) as CD_CD_CustId,
From LLA_Trx
Group By Week;
```

図４Ｂのテーブル４１０に図示される、結果として得られたテーブルの一覧では、各週に店舗のいずれかに来店した個別の客の数が不正確となるだろう。同様に、テーブルＬＬＡ＿ＴＲＸを集計するためにＳＵＭ関数が用いられるとする。

【００３０】

```
Create Table LLA_TRX_WEEK as
Select      Week,
      Sum(distinct CD_CustId) as Sum_CD_CustId,
From LLA_Trx
Group By Week;
```

すると、図４Ｃに図示される、結果として得られたテーブルの一覧でもまた、各週に店舗のいずれかに来店した個別の客の数が不正確になるだろう。例えば、週２の個別の客の数は、図３Ｃのテーブル３２０に基づくと４であり（客ＩＤ１，２，３および４）、図４Ｂのテーブル４１０の一覧では２であり、図４Ｃのテーブル４２０の一覧では５である。

【００３１】

前述したように、これは、個別の値の集計が非加法であるからである。週ごとの個別の客の数について正確な値を特定するためには、非加法集計の照会に対してはＬＬＡ＿ＴＲＸテーブルを無効にし、元のデータテーブルへ戻る必要がある。

【００３２】

開示された実施形態にしたがって生成されたＥＭＶデータ構造は、各グループセットの個別の値を、集計できるようなやり方で効率良く保存することによって、この問題を解決する。図２Ａにもどり、ステップ２０２では、一または複数のグループセットを特定し、ＥＭＶデータ構造を生成するために使用される新たな関数やその他の同様のコマンドを備えた、特別な照会が受領できる。ステップ２０３では、演算システムが、一または複数のエンティティマップベクトルを生成できる。もちろん、ステップ２０１、２０２、および２０３は、異なる順序で実行できる。例えば、照会をまず受領し（ステップ２０２）、その後、トークン化されたデータカラムが保存され（ステップ２０１）、そして、一または複数のエンティティマップベクトルが生成できる（ステップ２０３）。

【００３３】

ステップ２０２～２０３を容易にするために、値ではなくＥＭＶを各グループセットについて作成するよう演算システムに指示する、新たなＳＱＬ関数count(Additive_distinc

t<カラムの名前>）を定義することができる。ＥＭＶは、異なるエンティティに対して各ビットがマッピングされたビットマップベクトルでありうる。ＥＭＶのグループセット中に個別の値に関連する情報を保存することにより、エンティティと称される各個別の値をビットマップベクトルの一つのビットにマッピングすることができる。したがって、図３Ｃのトークン化された取引テーブル（Ｔｒｘ）３２０の上記の例を使用して、ＬＬＡテーブルＬＬＡ＿ＴＲＸを生成するために下記の定義を用いることができる。

【００３４】

```
Create Table LLA_TRX as
Select      Week,
      StoreId,
      Count(Additive_distinct CustId) as CD_CustId,
From Trx
Group By Week, StoreId;
```

図５Ａは、前記の結果得られるＬＬＡ＿ＴＲＸテーブル５００を示す。ＬＬＡ＿ＴＲＸテーブルのように、各グループセットに対応しているタプルＩＤ５０１、週５０２、および店舗ＩＤ５０３が保存される。ただし、個別の客のカウントを数字で保存するのではなく、各ビットが異なるＥＩＤを参照し、ビットの値がグループセット中のＥＩＤの有無を示すＥＭＶに、各グループセットについての個別の客が保存される。

【００３５】

もちろん、Additive_distinctのようなＳＱＬ拡張子が、各グループセットについてＥＭＶを作成するよう演算システムに指示する必要が無い。例えば、データベースエンジンが、Count (distinct<カラムの名前>)などの個別のデータ値の集計関数を含むカラムを備えたテーブルの作成を検出し、個別のデータ値のカウントを保存するためにＥＭＶを自動的に造ることができる。これによって、既存のフロントエンドツールでＥＭＶ構造が利用できるようにする。

【００３６】

グループセット（週１、店舗ＩＤ２）に対応する第二のＥＭＶ５１３について、以下にＥＭＶの構造を記載する。図５Ｂを参照すると、空のＥＭＶ５０５が示されている。先頭文字５０６の役割は、ＥＭＶにビットマップベクトルであると標識をつけることのみであり、それに適していればいかなる文字でもよく、また、省略することもできる。ＥＭＶ５０５の長さ寸法は、ビットスロットの数であり、先頭文字を含まない。ＥＭＶの長さはＥＭＶを生成するために用いられるカラムのドメイン中のＥＩＤの総数に等しい。換言すれば、ＥＭＶの長さは、ＥＭＶを生成するために用いられているドメイン中の固有の値の数に等しい。したがって、例えば、図５Ａのテーブル５００用にＥＭＶを生成するために用いられたドメインは、D_CustIdドメインであり、図３Ｂより、D_CustId ドメインは４つのＥＩＤを有していることがわかる。したがってそのドメインについてのＥＭＶの長さは、４という長さであり、この４はビットスロットに対応する。

【００３７】

再度図５Ｂを参照すると、ビットスロットの各々は、異なるＥＩＤに、ＥＩＤが割り当てられる順序で対応している。したがって、第一スロットはＥＩＤ＝１に対応し、第二スロットはＥＩＤ＝２に対応するなどである。ＥＭＶ中のビットスロットの各々のビットの値を特定するために、ＥＭＶに対応するグループセットを調べ、ＥＩＤ値を含むかどうかを確認する。したがって、ＥＭＶ５１３の第一ビットについては、グループセット（週１、店舗２）がCustId=1を含むかどうか、ステップ５０７で判断される。図３Ｃのテーブル３２０から、週１の間に店舗２にCustId=1が現れていないことがわかる。したがって、第一ビットの値は０である。

【００３８】

同じやり方で、ＥＭＶ５１３の第二ビットについては、グループセット（週１、店舗２）がCustId=2を含むかどうかステップ５０８で判断される。CustId=2 も週１の間に店舗２に現れていないので、第二ビットの値もゼロである。ＥＭＶ５１３の第三ビットについ

ては、グループセット（週１、店舗２）がCustId=3を含むかどうかステップ５０９で判断される。テーブル３２０から、週１の間に店舗２にCustId=3が２度現れていることがわかる。したがって、第三ビットの値は１に等しい。ＥＭＶ５１３中の第四ビットについては、グループセット（週１、店舗２）がCustId=4を含むかどうかステップ５１０で判断される。CustId=4もまた週１の間に店舗２に現れているので、第四ビットも同様に１に設定される。その結果得られたＥＭＶ５１３は、店舗２に週１の間に来店した個別の客を、加法集計を可能にするフォーマットで、把握している。

【００３９】

もしユーザがグループセットごとの個別の客の数を特定したければ、そのグループセットについてＥＭＶ中の各ビットを合計しさえすればよい。例えば、グループセット（週１、店舗１）についてのＥＭＶはＯＢ１１００であり、個別の客の総数は１＋１＋０＋０＝２である。グループセット（週２、店舗１）についてのＥＭＶはＯＢ１１０１であるので、そのグループセットの個別の客の総数は１＋１＋０＋１＝３である。

【００４０】

図２Ｂを参照し、以下にトークン化されていないデータカラムからＥＭＶを生成する方法を記載する。ステップ２１０では、図２Ａのステップ２０２と同様に、一または複数のグループセットを特定し、新たな拡張子またはその他の同様のコマンドを含み、データベースエンジンにＥＭＶデータ構造を生成するよう指示する、照会が受領される。ステップ２１１で、データのカラムに対応するドメイン中の固有データ値の語彙順に基づいて、グループセットの各々についてエンティティマップベクトルが生成される。図５Ｃおよび５Ｄは、図３Ａのトークン化されていないテーブル３００中のグループセット（週２０１００１、店舗１００２）について生成されたＥＭＶに関して、このプロセスをより詳細に示す。テーブル５２０は、客ＩＤドメインの、語彙順に配置された固有データ値を示す。図５Ｄに示されたＥＭＶ生成プロセスは、ＥＩＤ生成のステップをバイパスするためにこの語彙順序を使用する。空のＥＭＶ５２５に示されるように、生成されたＥＭＶの各々の長さは、ドメイン中の固有データ値の数に等しい。この場合、客ＩＤドメインには４つの固有データ値があるので、あるグループセットについて生成されたＥＭＶの長さは４となるであろう。各ＥＭＶ中の各ビットの位置は、固有データ値が語彙順に配列された一覧５２０中の、対応する固有データ値の語彙順位置に対応し、ＥＭＶ中の各ビットの値はグループセット中の対応する固有データ値の有無を示す。したがって、ＥＭＶ５３３中の第一ビットについては、グループセット（週２０１００１、店舗１００２）が語彙順で最も若い位置の客ＩＤを含むかどうかが判断される。この客ＩＤは１６５であり、図３Ａのテーブル３００から、この客ＩＤがグループセット（週２０１００１、店舗１００２）には出現しないことがわかるので、ビットはゼロとなる。同様に、ＥＭＶ５３３の第二ビットは語彙順で二番目に若い位置の固有データ値、１６７を参照し、これもグループセットには出現しないので、このビットはゼロとなる。ＥＭＶ５３３の第三ビットは語彙順で酸番目に若い位置の固有データ値、２６８を参照し、図３Ａのテーブル３００から、この値はグループセット（週２０１００１、店舗１００２）に出現しているとわかるので、第三ビットは１に設定される。同様に、第四ビットに対応する値、３０１、もグループセットに出現するので、第四ビットは１に設定される。この結果、図５Ｂに生成されたものと同じＥＭＶとなるが、これはトークン化されていないデータカラムから生成できる。

【００４１】

エンティティマップベクトルを利用しないＬＬＡテーブルの前記の例とは異なり、ＥＭＶを備えたＬＬＡテーブルは、ロールアッププロセス中に正確に集計できる。例えば、いずれかの店舗に来店した個別の客の数についての上位テーブルを、下記の定義を用いて、ＬＬＡテーブルから週でグループ化して生成することができる。

【００４２】

```
Create Table LLA_TRX_WEEK as
Select      Week,
      Count(Distinct CD_CustId) as EMV_CD_CustId,
```

```
From LLA_Trx
Group By Week;
```

図６Ａを参照すると、店舗にまたがるビットベクトルを集計し、週でグループ化するプロセスが示されている。ステップ６０１Ａ、６０１Ｂ、および６０１Ｃにおいて、集計されなければならないグループセットの各々が、各週について判断される。例えば、週１について合成すべきグループセットは、週１、店舗１と、週１、店舗２のグループセットに対応する客ＩＤである。週２について合成すべきグループセットは、週２、店舗１と、週２、店舗２と、週２、店舗３のグループセットに対応する客ＩＤである。週３について合成すべきグループセットは、週３、店舗１と、週３、店舗３のグループセットに対応する客ＩＤである。

【００４３】

ステップ６０２Ａ、６０２Ｂ、および６０２Ｃを参照すると、各週について合成すべきグループセットの各々のＥＭＶが、ブールＯＲ演算（Boolean OR）を使用して集計される。したがって、例えば、もし二つのＥＭＶをブールＯＲ演算で合成し、第一ＥＭＶのみが第一位置に「１」を有していれば、その結果得られるＥＭＶは「１」を第一位置に有する。

【００４４】

その結果得られた合成ＥＭＶがステップ６０３Ａ、６０３Ｂ、および６０３Ｃに示されている。各ＥＭＶは、異なる週のグループセットに対応する。前述のように、各週の個別の客のカウントは、ＥＭＶの各々のビットをすべて合計することにより特定できる。

【００４５】

図６Ｂは、その結果得られたテーブル６００を示し、テーブル６００は図６Ａの合成ＥＭＶに基づいて、各グループセットについてのタプルＩＤ６０１、週番号６０２、および個別の客のカウント６０３を備えている。図３Ｃの元のテーブル３２０を参照すると、テーブル６００の個別の客のカウント６０３は、正確に示すｔｈａｔ４人の個別の客（１，２，３、および４）が週１にいずれかの店舗に来店し、４人の個別の客（１，２，３、および４）が週２にいずれかの店舗に来店し、２人の個別の客（１および２）が週３にいずれかの店舗に来店したことが、わかる。

【００４６】

もちろん、非加法集計関数の集計は、グループセット中の個別の要素の数をカウントすることに限定されない。ユーザが、あるグループセット中の個別の値のセットを合計したい、またはグループセット中の個別の値のセットの平均値を求めたいなら、ルックアップ辞書中のＥＭＶの各ビットに対応する固有データ値を探索し、加算して合計したり、またはもしビットが「１」であれば平均値に加えたりすることによって、達成できる。さらに、カスタム化計算やメトリクスなど、他の目的のためにエンティティマップベクトル中のビットに対応する固有データ値の各々を検索するために、ルックアップ辞書を用いることができる。エンティティマップベクトルが、固有データ値の語彙順の配列に基づいて、トークン化されていないデータ値から生成される状況では、同じ語彙順配列関係を、各ビットに対応する固有データ値を検索するために用いることができる。例えば、ＥＭＶの第一ビットが「１」なら、最も若い語彙順位置の固有データ値を、セット中のデータ値を含む、合計、平均、または他の計算に組み入れることができる。同様に、ＥＭＶの第二ビットが「０」なら、二番目に若い語彙順位置に対応する固有データ値は、合計、平均、または他の計算から除外することができる。

【００４７】

パーティション化されたデータベース中の照会処理およびパーティション化されたデータベース中のデータの表現に基づいて、ＥＭＶのプロセスの大要を以下に記載する。取引テーブルを含む３つのパーティションデータベースを仮定すると、そのテーブルの論理ビューは、Viewとして下記のようにあらわされる。

```
Create View V1 as
Select * from Partition1.TRX
```

```
Union All
Select * from Partition2.TRX
Union All
Select * from Partition3.TRX
```

テーブルのパーティション化は、要求の効率的な分散並列処理を可能にするが、非加法集計関数は例外である。例えば、図３Ａのテーブル３００と同様のＴＲＸという名の取引テーブルの元の照会が、「Select Week, Sum(Spent) from TRX group by Week（週を選択、週によるＴＲＸグループからの（出費額）を合計）」であるが、テーブルが３つのパーティションに分散されていれば、照会は、分散された要求に変形することができる。

```
Select Week, Sum(P_Spent) from (
      Select Week, Sum(Spent) as P_Spent from Partition1.TRX group by Week
      Union All
      Select Week, Sum(Spent) as P_Spent from Partition2.TRX group by Week
      Union All
      Select Week, Sum(Spent) as P_Spent from Partition3.TRX group by Week
      ) T
Group By Week
```

その分散された要求については、「Select Week, Sum(Spent) as P_Spent from Partition1.TRX group by Week（週を選択、週によるパーティション１．ＴＲＸグループからのＰ＿Ｓｐｅｎｔとして（出費額）を合計）」のような、各照会セグメントが独立して実行でき、その中間結果セットをまとめて付加して最後に再処理し、最終的な照会結果を作成できる。

【００４８】

「Select Week, Count(Distinct CustId) from TRX group by Week（週を選択、週によるＴＲＸグループからの（個別の客ＩＤ）をカウント）」のような非加法集計関数要求の場合、この要求は分散された要求に変形できる。

```
Select Week, Count(Distinct CustId) from (
            Select Week, CustId from Partition1.TRX group by Week, CustId
            Union All
            Select Week, CustId from Partition2.TRX group by Week, CustId
            Union All
            Select Week, CustId from Partition3.Trx group by Week, CustId
            ) T
      Group By Week
```

分散された要求はこの場合、各パーティションから固有のＣｕｓｔＩｄの一覧を保存するが、膨大なデータをパーティションの各々の間で交換し、その後処理するよう命ずることがありうる。例えば、第一週についての第一パーティション中のＣｕｓｔＩｄ（客ＩＤ）の一覧を、第一週についての第二パーティション中のＣｕｓｔＩｄ（客ＩＤ）の一覧と比較し、重複が無いか判断しなければならないだろう。その結果、かかる分散された要求は、要求実行プロセスに支障をきたす。

【００４９】

パーティション間で交換されるデータ量を削減し性能を向上させるために、非加法集計関数を加法関数に変形する方法を一または複数のパーティション化されたデータベースに適用することができる。例えば、元の照会「Select Week, Count(Distinct CustId) from TRX group by Week（週を選択、週によるＴＲＸグループからの（個別の客ＩＤ）をカウント）」が下記のように変形できる。

```
Select Week, Count(Distinct CustId) from
(
      Select Week, Count(Additive_distinct CustId) CustId from Partition1.TRX gr
oup by Week
```

```
        Union All
    Select Week, Count(Additive_distinct CustId) CustId  from Partition2.TRX g
roup by Week
        Union All
    Select Week, Count(Additive_distinct CustId) CustId from Partition3.Trx gr
oup by Week
    ) T
    Group By Week
```

図７Ａは、開示された実施形態にかかる、この分散された要求を処理するステップを示す。３つの第一ステップ７０１は、パーティションの各々に対して平行して行うことができ、各パーティションに適用される図２ＡにＥＭＶ生成ステップと同様である。もちろん、これらステップは平行して行われなくてもよく、なんらかの適した順序で行える。前記３つの第一ステップ７０１がパーティションの各々について実行され、その結果、各パーティションに一つずつ、計３つのＥＭＶセットが得られ、ＥＭＶの各桁はローカルパーティションＥＩＤを参照している。例えば、図８Ａは、３つのパーティションに対応して、３つの仮想ＥＩＤマッピングテーブル８０１Ａ、８０１Ｂ、および８０１Ｃと共に、３つの仮想ＥＭＶセット８０２Ａ、８０２Ｂ、および８０２Ｃを、パーティションの各々にある固有データ値の各々について示している。マッピングテーブル８０１Ａ～８０１Ｃ中のＥＩＤはすべて異なる固有データ値に対応しているので、また、各パーティション中のＥＩＤの数が様々であるので、ＥＭＶの長さも、ＥＭＶセット８０２Ａ～８０２Ｃ中の各ＥＭＶのビットの意味も、様々である。その結果、ＥＭＶセット８０２Ａ～８０２Ｃは、このままではｕｎｉｏｎ演算子によって合成することができない。

【００５０】

図７Ａを参照すると、ステップ７０２で、ドメインパーティションの各々からグローバルドメインを構築するためにドメイン結合プロセスをおこない、ＥＭＶセットを標準化できる。図７Ｂへ続き、ステップ７０３で、グローバルドメイン中の固有データ値の各々について、グローバルＥＩＤが生成される。グローバルＥＩＤがパーティションＥＩＤとしての同じ固有データ値に対応するので、図７Ｂのステップ７０４でパーティションＥＩＤの各々がグローバルＥＩＤに変換できる。ステップ７０５では、パーティションに特定のＥＭＶがグローバルＥＭＶに変換され、そのすべてが各ビットにおいて、同じ長さを有し、同じ固有データ値を参照する。これはパーティション－グローバルＥＩＤマッピングテーブルを使用して、パーティションに作り出されたＥＭＶの各々を再構築し、グローバルＥＩＤに対してマッピングする新たなＥＭＶを作り上げることにより達成される。

【００５１】

当該プロセスを示すために、図８Ｂは、図８Ａのドメインパーティションのドメイン結合の結果得られたグローバルドメイン８０３と、対応するグローバルＥＩＤ８０４とを示す。図８Ｃは、パーティション１、８１１、パーティション２、８１２およびパーティション３、８１３用のパーティション－グローバルＥＩＤマッピングテーブルを示す。前述のように、パーティション－グローバルＥＩＤマッピングは、パーティションＥＩＤとグローバルＥＩＤとが固有データ値を相互参照することによって生成される。マッピングテーブルはその後、パーティション特定ＥＭＶをグローバルＥＭＶに変換するために用いられる。変換の結果生成されたグローバルＥＭＶは、パーティション１、８２１、パーティション２、８２２、およびパーティション３、８２３について図８Ｄに示されている。

【００５２】

パーティション特定ＥＭＶがグローバルＥＭＶに変換されると、前述のようにブールＯＲ演算で集計できる。その結果得られた合成グローバルＥＭＶは、合成パーティションすべてにわたるあるグループセットにおいて、固有データ値が有るかどうかを示すことになろう。さらに、合成グローバルＥＭＶ中のビットの合計は、パーティションにわたってそのＥＭＶに対応するグループセット中の個別のデータ値の総数と等しくなる。

【００５３】

このドメイン結合プロセスは、データベース更新演算に利用できる。データウェアハウス環境では、更新プロセスが新たなデータを付加する工程を含むことがよくある。関連付けられたパーティションＬＬＡテーブルを生成するために新たなパーティションとして付加されたデータを処理し、上述した技術を使用してパーティションＬＬＡテーブルを既存のＬＬＡテーブルと合成することにより、前記新たなデータを既存のＬＬＡテーブルと結合することができる。

【００５４】

例えば、パーティションの各々についてのグローバルＥＭＶが、データキューブに保存される。図９は、開示された実施形態にかかる、グローバルＥＩＤを保存するデータキューブ９００の例を示す。図示されているように、次元には、ＥＭＶ次元９０１、グループセット次元９０２、およびパーティション次元９０３がある。新たなデータ更新が受領されると、パーティションとして扱い、前述のドメイン結合マージプロセスを使用してデータキューブに加えることができる。

【００５５】

前述のように、ＥＩＤも、データ値の語彙順ではなく、データ値が出現した経時的な順序で割り当てられてもよい。このようにＥＩＤを割り当てることにより、データウェアハウスの更新のためのドメイン結合ステップを省略できる。図１０を参照すると、既存のデータウェアハウスが１００１に示されている。この例では、データ値が処理された経時的な順序を用いてＥＩＤを各データ値に割り当てている。テーブル１００２は、以下の順序で処理された４つのデータ値用のＥＩＤを示している：３６８、１６７、１６５、２６８。さらに、テーブル１００３は、データ値を含む３つのグループセットについてのＥＭＶを示している。したがって、例えば、第二のＥＭＶ「ＯＢ１１１０」は、データ値３６８、１６７、および１６５を含むグループセットに対応する。

【００５６】

図１０はまた、データウェアハウスのデータ更新１００４を示す。データ更新は、上述したパーティションと同様である。ローカルＥＩＤマッピングテーブル１００５は、同様に経時的順序で割り当てられた、更新におけるデータ値についてのＥＩＤを示している。さらに、二つのグループセットについてのＥＭＶが１００６に示されている。この場合、更新におけるデータ値のうち二つは新規であり、データ値の一つはすでにデータウェアハウスに存在していたものである。

【００５７】

更新されたデータウェアハウスが１００７に示されている。ＥＩＤが経時的順序で割り当てられているので、ドメイン結合プロセスを行う必要がない。データ更新における二つの新たなデータ値、３９２および１６３が、次に使用可能な二つのＥＩＤ、この場合５および６、に割り当てられる。既存のＥＩＤの再マッピングが無いので、各ＥＭＶの最後に新たなＥＩＤの数と等しい数のゼロを加えることによって各ＥＭＶの長さを伸ばすだけで、ＥＭＶの既存のテーブル１００３が更新される。この結果、更新されたＥＭＶを備えたテーブル１００９となる。もちろん、ある位置に文字が無ければゼロと解釈することができるので、ゼロが必ずしもＥＭＶに付加されなくてもよい。

【００５８】

データ値の経時的な順序に基づいてＥＩＤが割り当てられている場合にデータ更新に必要な再マッピングのみが、実際のデータ更新においてＥＭＶに対して行われる。この場合、更新１００６のＥＭＶは、テーブル１０１０の更新されたＥＭＶを生成するためにはローカルＥＩＤテーブル１００５ではなくグローバルＥＩＤテーブル１００８を使用するために、再マッピングされなければならない。したがって、例えば、更新１００６の第一ＥＭＶが、値３９２および１６３（ＥＭＶの一番目および三番目の位置の「１」）を含むグループセットに対応し、これらデータ値に対応するグローバルＥＩＤが５および６なので、データ更新内の第一ＥＭＶに対応する更新されたデータテーブル１０１０中の第一ＥＭＶは、五番目および六番目の位置に「１」を有することになろう。

【００５９】

パーティションに特定された一または複数のＥＭＶが第一データパーティション中のトークン化されていないデータカラムから生成され、一または複数の他のパーティションと結合されるというシナリオでは、第一データパーティション中のトークン化されていないデータカラムに対応する第一ドメインパーティションを、グローバルドメインを生成するためのドメインパーティションの群の中の、一または複数の他のドメインパーティションと合成できる。ＥＭＶ生成のプロセスと同様に、グローバルドメイン中のすべての固有データ値の語彙順配列に基づいて、パーティションに特定された一または複数のＥＭＶに対応する一または複数のグローバルＥＭＶを生成するために、このグローバルドメインを用いることができる。各グローバルＥＭＶの長さはグローバルドメイン中の固有データ値の数と等しくてもよく、グローバルＥＭＶ中の各ビットの位置はグローバルドメイン中の固有データ値の語彙順の一覧における固有データ値に対応する語彙順の位置に対応してもよく、グローバルＥＭＶ中の各ビットの値は、グループセット中における、対応する固有データ値の有無を示すことができる。

【００６０】

前述のドメインマージプロセスをバイパスするために使用することができ、分散システムでのデータベースの演算をより効率的にできる、ドメイン同期プロセスを、下記に記載する。図１１Ａは、分散データベース中の二つのパーティション、１１０１および１１０２を示す。図８Ａの分散データベースと同様に、パーティションの各々はローカルＥＩＤテーブルを含み、パーティション１１０１にはテーブル１１０４、パーティション１１０２にはテーブル１１０６を備える。この例では、ＥＩＤはデータ値の語彙順に基づいて割り当てられているものとして示されているが、ＥＩＤは前述のように経時的に割り当てられられてもよい。さらに、分散データベースは、グローバルＥＩＤテーブル１１０３を備える。グローバルＥＩＤテーブル１１０３には、データ値すべてをグローバルＥＩＤにマッピングしたものが入っている。繰り返しになるが、これらグローバルＥＩＤはデータ値の語彙順に基づいて割り当てられているが、データ値の経時的な順序にもとづいて割り当てることもできる。グローバルＥＩＤテーブル１１０３はパーティション１１０１およびパーティション１１０２の外に保存されているものとして示されているが、グローバルＥＩＤテーブルは、パーティションの一つ、または両方に保存されることも可能である。

【００６１】

各パーティション１１０１および１１０２もまた、ローカルＥＩＤ－グローバルＥＩＤマッピングテーブル、パーティション１１０１についてのテーブル１１０５、およびパーティション１１０２についてのテーブル１１０７を含む。ＥＭＶ１１０８および１１０９の例を使用して下記に記載するように、これらマッピングテーブルは、ローカルＥＭＶをグローバルＥＭＶに変えるために使用できる。

【００６２】

パーティション１１０１またはパーティション１１０２のどちらかで更新が受領されると、新たな固有データ値がグローバルＥＩＤテーブル１１０３へ送られる。これら新たな固有データ値は、グローバルＥＩＤテーブルを更新するために用いることができる。例えば、ＥＩＤが固有データ値の語彙順で割り当てられる時、グローバルＥＩＤテーブル１１０３中のデータ値が再ソートされ、受領された新たな固有データ値を考慮してグローバルＥＩＤが固有データ値の各々に再割り当てされる。経時的に割り当てられたＥＩＤが用いられるなら、新たな固有データ値が次に使用可能なＥＩＤに割り当てられて、既存の固有データ値のすべてに対してＥＩＤを再割り当てる必要がないので、このプロセスは大幅に簡略化される。

【００６３】

グローバルＥＩＤテーブル１１０３が更新された後、新たなグローバルＥＩＤがパーティション１１０１および１１０２へ戻され得る。新たなグローバルＥＩＤを用いて、ローカルＥＩＤ－グローバルＥＩＤマッピングテーブル１１０５および１１０７の各々を更新することができる。繰り返しになるが、このプロセスには経時的に割り当てられたＥＩＤが効率的である。新たなグローバルＥＩＤのみが新たな固有データ値について割り当てら

れるからである。語彙順で割り当てられたＥＩＤでは、既存の固有データ値についてのすべてのＥＩＤが再シャッフルされうる。

【００６４】

照会が複数のパーティションからＥＭＶを求める場合、ローカルＥＩＤ－グローバルＥＩＤマッピングテーブル、１１０５および１１０７、は、なんらかのローカルＥＭＶをグローバルＥＭＶへ変換するために用いることができる。図１１Ｂを参照すると、パーティション１１０１中の見本ＥＭＶ１１０８についての変換プロセスが示されている。図１１Ｂに示されるように、ローカルＥＭＶ中の各ビットは、そのローカル位置から、等価のグローバルＥＭＶ１１１０中のそのグローバル位置へ、マッピングされる。したがって、例えば、ローカルＥＩＤ－グローバルＥＩＤマッピングテーブル１１０５がローカルＥＩＤ「３」をグローバルＥＩＤ「５」へマッピングするので、ローカルＥＭＶ１１０８中の三番目の位置ビットの値は、グローバルＥＭＶ１１１０の五番目の位置へ割り当てられる。グローバルＥＭＶ１１１０中の四番目の位置など、グローバルＥＭＶ中の位置へのマッピングが無ければ、その値は、グループセット中のある固有データ値が無いことを反映してゼロに設定される。なぜなら、もしその値が存在すれば、ローカルＥＩＤ－グローバルＥＩＤマッピングテーブル１１０５中にそれがあったはずだからである。同様に、パーティション１１０２中のローカルＥＭＶ１１０９からグローバルＥＭＶ１１１１を生成するためには、ローカルＥＩＤ－グローバルＥＩＤマッピングテーブル１１０７を使用できる。

【００６５】

このドメイン同期プロセスを通じて、上述されたドメインマージ手順をバイパスすることができ、その結果処理時間が改善でき、一度にエクスポートする必要のある固有データ値の数を減らすことができる。もちろん、各パーティションにＥＭＶが一つ示されているのは、明瞭化だけのためである。実際は、あるパーティションにローカルＥＩＤおよび値が存在すれば、いくつかのグループセットおよびＥＭＶにその値が入るはずである。さらに、二つのパーティションが示されているのも単に明瞭化のためだけであり、データベースが含むパーティションの数はどんな数であってもよい。

【００６６】

上記の技術の一または複数は、一または複数のコンピュータシステム中に実現することができ、または、それらで構成されることができる。図１２は、コンピューティング環境１２００の一般例を示す。コンピューティング環境１２００は、記載された実施形態の用途または機能の範囲についていかなる限定も示唆するものではない。

【００６７】

図１２を参照すると、コンピューティング環境１２００は、少なくとも一つの処理ユニット１２１０およびメモリ１２２０を含む。処理ユニット１２１０はコンピュータ実行可能な 命令を実行し、現実のまたは仮想のプロセッサであってもよい。多重処理システムにおいては、多数の処理ユニットがコンピュータ実行可能な命令を実行して処理パワーを増加させている。メモリ１２２０は、一時的なメモリ（例えば、レジスタ、キャッシュ、ＲＡＭなど）、非一時的なメモリ（例えば、ＲＯＭ、ＥＥＰＲＯＭ、フラッシュメモリなど）、またはその二つの組合せであってもよい。メモリ１２２０は、上記技術を実現するソフトウェア１２８０を記憶していてもよい。

【００６８】

コンピューティング環境は、さらに追加の構成を有することができる。例えば、コンピューティング環境１２００は、記憶装置１２４０、一または複数の入力装置１２５０、一または複数の出力装置１２６０、および一または複数の通信接続部１２９０を備える。バス、制御器、またはネットワークなどの相互接続機構１２７０は、コンピューティング環境１２００の構成部分を相互接続する。通常、オペレーティングシステムソフトウエアまたはファームウェア（図示省略）は、コンピューティング環境１２００で作動するその他のソフトウェア用の操作環境を提供し、コンピューティング環境１２００の構成部分の動作を調整する。

【００６９】

記憶装置１２４０は着脱可能であっても、着脱不可であってもよく、磁気ディスク、磁気テープ、カセット、ＣＤ－ＲＯＭ、ＣＤ－ＲＷ、ＤＶＤなど、情報を記憶するために用いることができ、コンピューティング環境１２００中でアクセスすることができる媒体を含む。記憶装置１２４０は、ソフトウェア１２８０への命令を記憶できる。

【００７０】

入力装置（複数も可）１２５０は、キーボード、マウス、ペン、トラックボール、タッチ画面、またはゲーム制御器などのタッチ入力装置、音声入力装置、走査装置、デジタルカメラ、遠隔制御、コンピューティング環境１２００に入力を行う他の装置でありうる。出力装置（複数も可）１２６０は、表示器、テレビ、モニタ、プリンタ、スピーカ、コンピューティング環境１２００からの出力を行う他の装置でありうる。

【００７１】

通信接続部（複数も可）１２９０は、通信媒体を介して別のコンピューティングエンティティへの通信を可能にする。通信媒体は、コンピュータ実行可能な命令、音声または画像情報、変調データ信号のデータ、等の情報を伝達する。変調データ信号は、信号の特徴の一つまたは複数が信号中の情報を暗号化するように設定または変更された信号である。例であって限定ではないが、通信媒体としては、電気、光学、ＲＦ、赤外線、音響などのキャリアによって実現される有線または無線技術などがある。

【００７２】

インプリメンテーション（実現）は、コンピュータ可読媒体の一般的なコンテキストで説明できる。コンピュータ可読媒体は、コンピューティング環境内でアクセスできる媒体であれば、いずれの媒体であってもよい。例であって限定ではないが、コンピューティング環境１２００内においては、コンピュータ可読媒体としては、メモリ１２２０、記憶装置１２４０、通信媒体、および上記のいずれかの組み合わせなどが挙げられる。

【００７３】

図１２は、コンピューティング環境１２００、表示装置１２６０、および入力装置１２５０を個別の装置として示しているが、これは識別のしやすさのためである。コンピューティング環境１２００、表示装置１２６０、および入力装置１２５０は個別の装置（例えば、モニタやマウスに有線で接続されたパーソナルコンピュータなど）でもよく、単一の装置に統合されたもの（例えば、スマートフォンやタブレットなどタッチディスプレイを備えたモバイル装置など）でありえ、または装置の組み合わせ（例えば、タッチ－スクリーン表示装置と動作的に連結される演算装置、単一の表示装置および入力装置に取り付けられた複数の演算装置など）でありえる。コンピューティング環境１２００はセットトップボックスや、パーソナルコンピュータ、または、一または複数のサーバであってよく、例えばネットワーク接続されたサーバの形式、クラスタ化サーバ環境、または演算装置のクラウドネットワークでありうる。

【００７４】

記載された実施形態を参照して本発明の原理を記載し説明したが、記載された実施形態は、構成や詳細においてかかる原理から逸脱することなく変更できることは、認識されるであろう。特段の記載の無い限り、本明細書に記載されたプログラム、プロセス、または方法は、特定の種別のコンピューティング環境に関連するわけではなく、限定もされないことを、理解されたい。各種の汎用のまたは特殊なコンピューティング環境は、本明細書に記載された教示に従って、使用されたり操作が実行されたりできる。ソフトウェアに示される本実施形態の要件は、ハードウェアで実現でき、その逆も可である。

【００７５】

本発明の原理が適用できる、考えられうる多くの実施形態を鑑み、以下の請求項および均等物の範囲および精神に該当するような実施形態はすべて、本発明であると主張する。

【図１Ａ】

顧客名	店舗 ID	出費額
Bill	1	10
Bill	2	5
George	1	11
George	2	11
Max	2	21
Larry	1	31
Larry	2	18

【図１Ｂ】

店舗 ID	来店数
1	3
2	4

【図１Ｃ】

店舗 ID	出費額
1	52
2	55

【図１Ｄ】

店舗 ID	平均出費額	来店数
1	17 1/3	3
2	13 3/4	4

【図１Ｅ】

店舗 ID	最小出費額
1	10
2	5

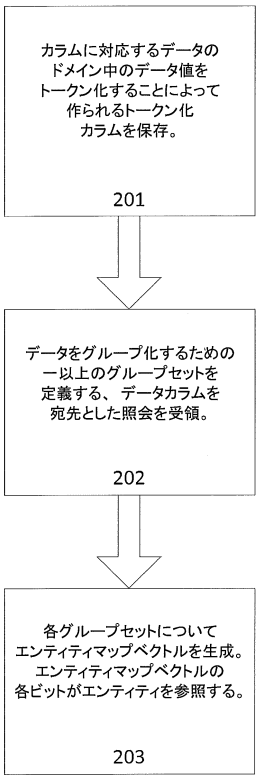
【図１Ｆ】

店舗 ID	最大出費額
1	31
2	21

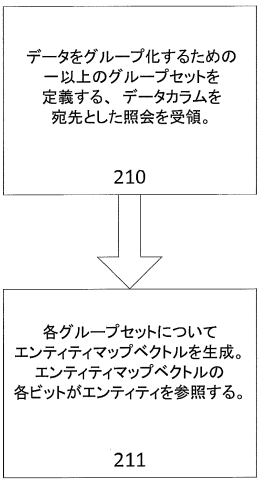
【図１Ｇ】

店舗 ID	個別の客の数
1	3
2	4

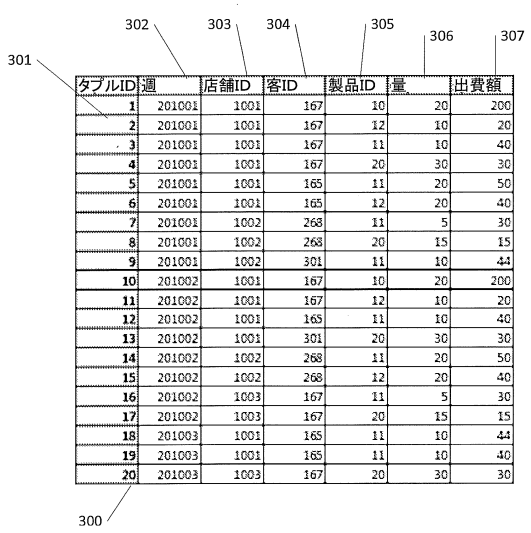
【図２Ａ】



【図２Ｂ】



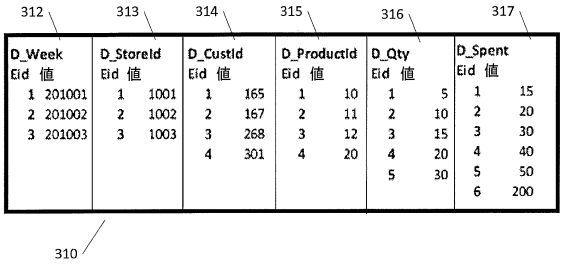
【図３Ａ】



301	302	303	304	305	306	307
タブルID	週	店舗ID	客ID	製品ID	量	出費額
1	201001	1001	167	10	20	200
2	201001	1001	167	12	10	20
3	201001	1001	167	11	10	40
4	201001	1001	167	20	30	30
5	201001	1001	165	11	20	50
6	201001	1001	165	12	20	40
7	201001	1002	268	11	5	30
8	201001	1002	268	20	15	15
9	201001	1002	301	11	10	44
10	201002	1001	167	10	20	200
11	201002	1001	167	12	10	20
12	201002	1001	165	11	10	40
13	201002	1001	301	20	30	30
14	201002	1002	268	11	20	50
15	201002	1002	268	12	20	40
16	201002	1003	167	11	5	30
17	201002	1003	167	20	15	15
18	201003	1001	165	11	10	44
19	201003	1001	165	11	10	40
20	201003	1003	167	20	30	30

300

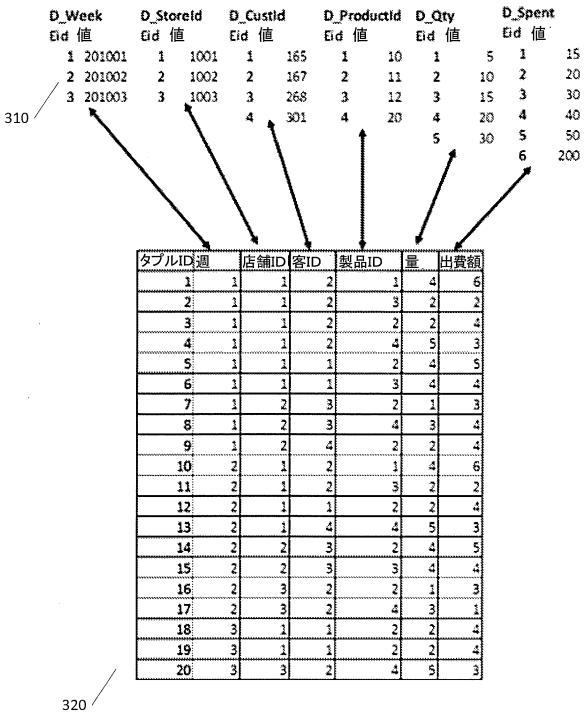
【図３Ｂ】



312	313	314	315	316	317
D_Week	D_StoreId	D_CustId	D_ProductId	D_Qty	D_Spent
Eid 値	Eid 値	Eid 値	Eid 値	Eid 値	Eid 値
1 201001	1 1001	1 165	1 10	1 5	1 15
2 201002	2 1002	2 167	2 11	2 10	2 20
3 201003	3 1003	3 268	3 12	3 15	3 30
		4 301	4 20	4 20	4 40
				5 30	5 50
					6 200

310

【図３Ｃ】



D_Week Eid 値: 1 201001; 2 201002; 3 201003

D_StoreId Eid 値: 1 1001; 2 1002; 3 1003

D_CustId Eid 値: 1 165; 2 167; 3 268; 4 301

D_ProductId Eid 値: 1 10; 2 11; 3 12; 4 20

D_Qty Eid 値: 1 5; 2 10; 3 15; 4 20; 5 30

D_Spent Eid 値: 1 15; 2 20; 3 30; 4 40; 5 50; 6 200

310

タブルID	週	店舗ID	客ID	製品ID	量	出費額
1	1	1	2	1	4	6
2	1	1	2	3	2	2
3	1	1	2	2	2	4
4	1	1	2	4	5	3
5	1	1	1	2	4	5
6	1	1	1	3	4	4
7	1	2	3	2	1	3
8	1	2	3	4	3	4
9	1	2	4	2	2	4
10	2	1	2	1	4	6
11	2	1	2	3	2	2
12	2	1	1	2	2	4
13	2	1	4	4	5	3
14	2	2	3	2	4	5
15	2	2	3	3	4	4
16	2	3	2	2	1	3
17	2	3	2	4	3	1
18	3	1	1	2	2	4
19	3	1	1	2	2	4
20	3	3	2	4	5	3

320

【図４Ａ】

タブルID	週	店舗ID	客ID
1	1	1	2
2	1	2	2
3	2	1	3
4	2	2	1
5	2	3	1
6	3	1	1
7	3	3	1

400

【図４Ｂ】

タブルID	週	CD_CD_CustId
1	1	1
2	2	2
3	3	1

410

【図４Ｃ】

タブルID	週	Sum_CD_CustId
1	1	4
2	2	5
3	3	2

420

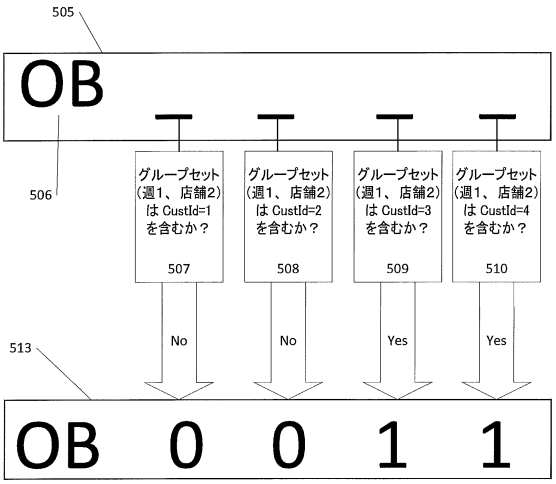
【図５Ａ】

501	502	503	504
タブルID	週	店舗ID	客ID
1	1	1	0B1100
2	1	2	0B0011
3	2	1	0B1101
4	2	2	0B0010
5	2	3	0B0100
6	3	1	0B1000
7	3	3	0B0100

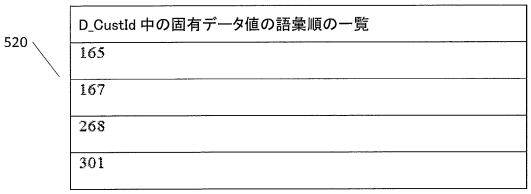
513

500

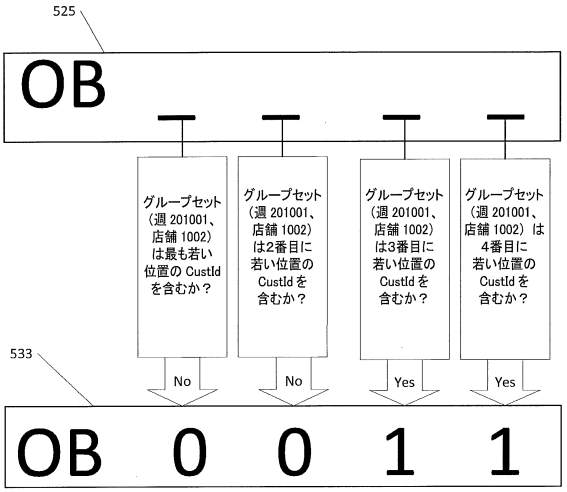
【図５Ｂ】



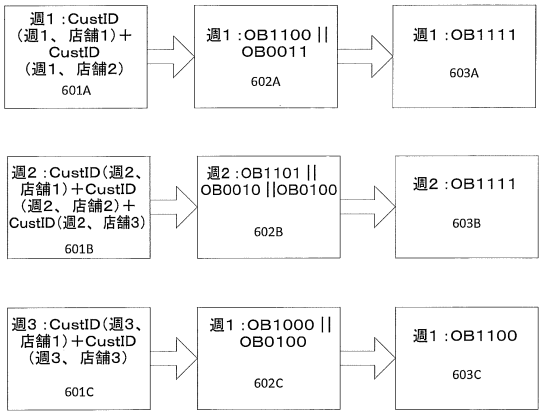
【図５Ｃ】



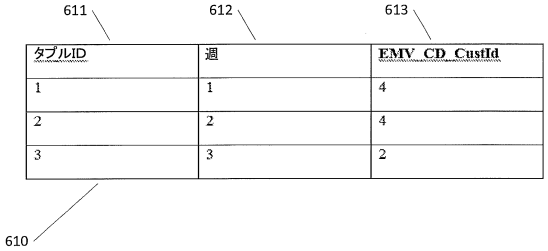
【図５Ｄ】



【図６Ａ】

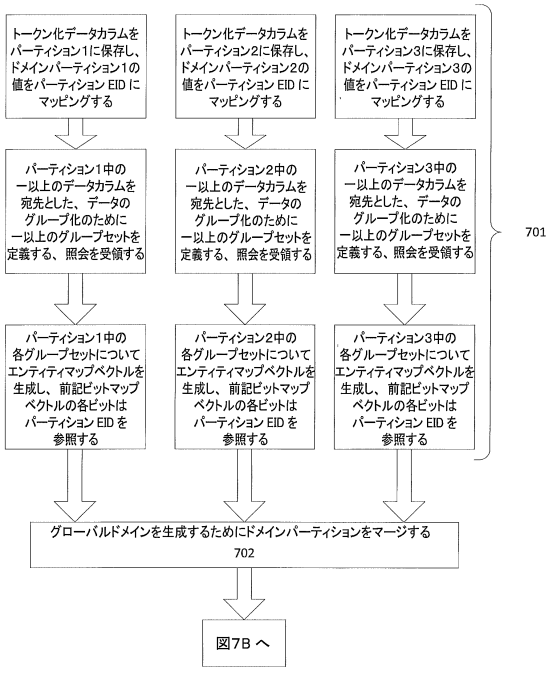


【図６Ｂ】

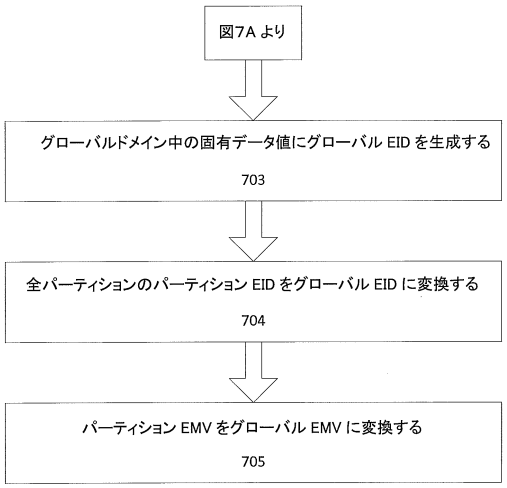


テーブルID	週	EMV_CD_CustId
1	1	4
2	2	4
3	3	2

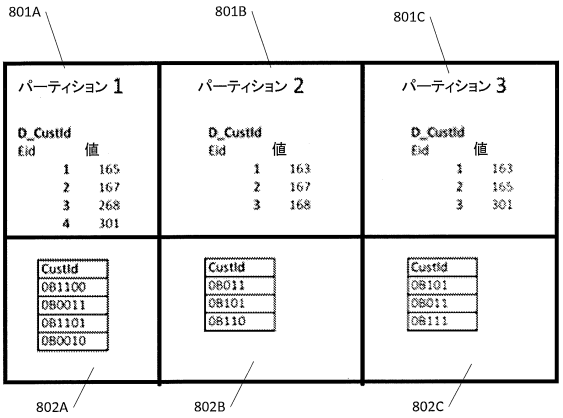
【図７Ａ】



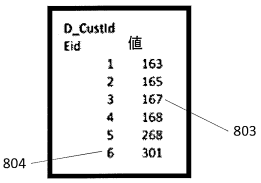
【図７Ｂ】



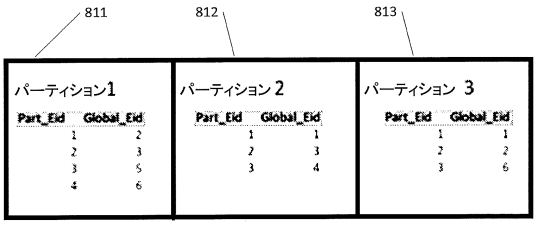
【図８Ａ】



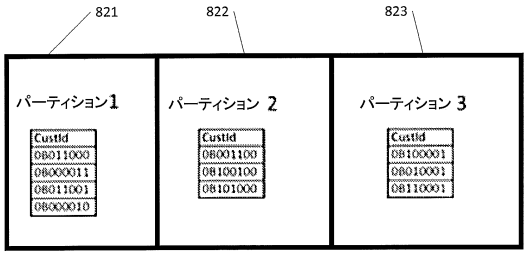
【図８Ｂ】



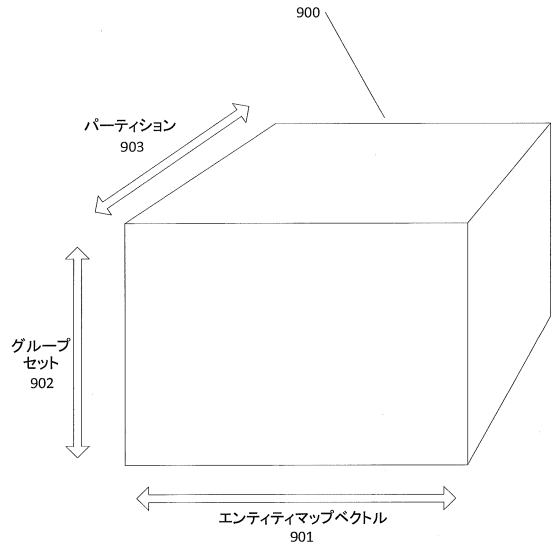
【図８Ｃ】



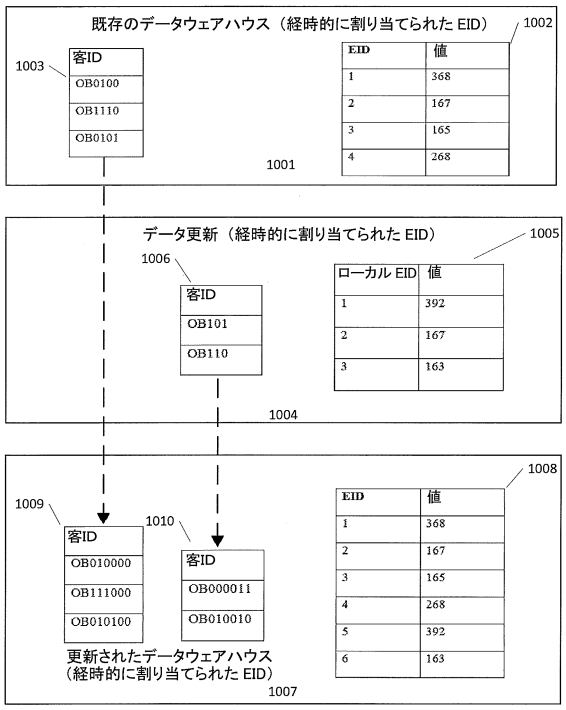
【図８Ｄ】



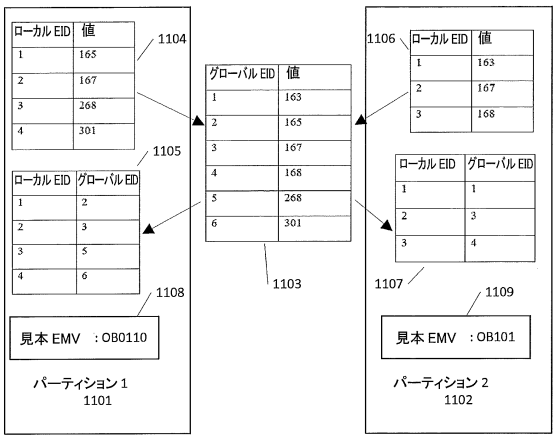
【図９】



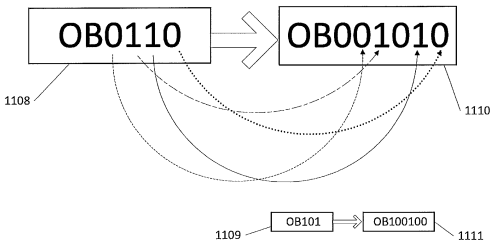
【図１０】



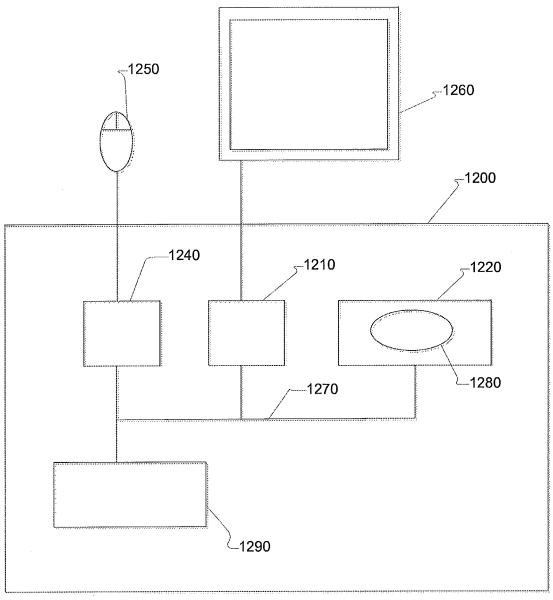
【図１１Ａ】



【図１１Ｂ】



【図１２】



フロントページの続き

(72)発明者　ファデエイチェフ、　ユージェニー
カナダ　エイチ４ブイ　３エヌ２、ケベック、モントリオール、プリンスオブウェールズ、５１３９

審査官　吉田　誠

(56)参考文献　特開平１０－９７５４４（ＪＰ，Ａ）
米国特許第６６４７３７２（ＵＳ，Ｂ１）
米国特許第７６５３６０５（ＵＳ，Ｂ１）
米国特許第９２１８３７９（ＵＳ，Ｂ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１７／３０