



(12) **Veröffentlichung**

der internationalen Anmeldung mit der
 (87) Veröffentlichungs-Nr.: **WO 2014/143791**
 in deutscher Übersetzung (Art. III § 8 Abs. 2 IntPatÜG)
 (21) Deutsches Aktenzeichen: **11 2014 001 361.7**
 (86) PCT-Aktenzeichen: **PCT/US2014/027907**
 (86) PCT-Anmeldetag: **14.03.2014**
 (87) PCT-Veröffentlichungstag: **18.09.2014**
 (43) Veröffentlichungstag der PCT Anmeldung
 in deutscher Übersetzung: **26.11.2015**

(51) Int Cl.: **G06F 17/16 (2006.01)**

(30) Unionspriorität:
13/835,590 **15.03.2013** **US**

(71) Anmelder:
INFORMATICA CORPORATION, Redwood City, Calif., US

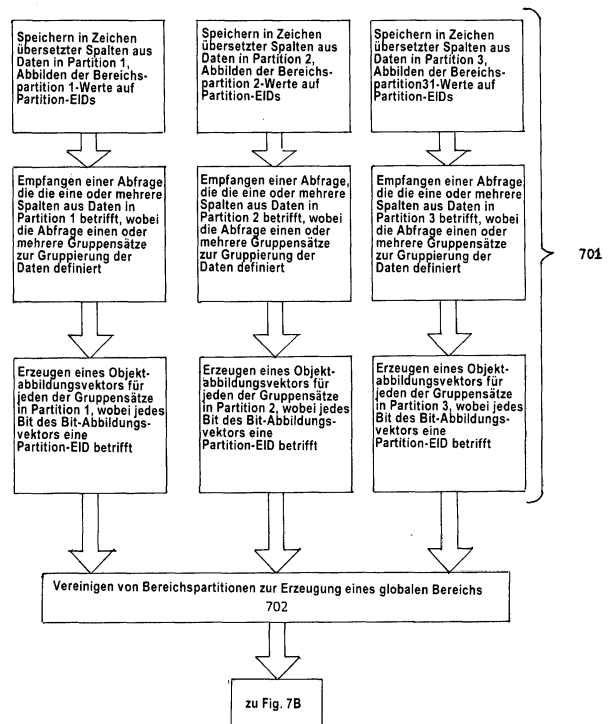
(74) Vertreter:
Grünecker Patent- und Rechtsanwälte PartG mbB, 80802 München, DE

(72) Erfinder:
Grondin, Richard, Ste-julie, QC, CA; Fadeitchev, Evgueni, Montreal, QC, CA

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

(54) Bezeichnung: **Verfahren, Vorrichtung und computerlesbares Medium für effiziente Ausführung von Operationen an individuellen Datenwerten**

(57) Zusammenfassung: Eine Vorrichtung, ein computerlesbares Medium und ein computerimplementiertes Verfahren zum effizienten Ausführung von Operationen an individuellen Datenwerten mit: Speichern einer in Zeichen übersetzten Spalte aus Daten in einer Tabelle durch Abbildung jedes eindeutigen Datenwertes in einem entsprechenden Bereich auf eine eindeutige Objekt-ID, und Ersetzen jedes Datenwertes in der Spalte durch die entsprechende Objekt-ID, um eine Spalte aus in Zeichen übersetzten Daten zu erzeugen, die eine oder mehrere Objekt-IDs enthalten, Empfangen einer Abfrage, die die Spalte aus Daten betrifft, wobei die Abfrage einen oder mehrere Gruppensätze zur Gruppierung der Daten definiert, die in Reaktion auf die Abfrage abgerufen werden, und Erzeugen eines Objektabbildungsvektors für jeden Gruppensatz, wobei die Länge jedes Objektabbildungsvektors gleich der Anzahl eindeutiger Objekt-IDs für den Bereich ist, und wobei der Wert jedes Bits in dem Objektabbildungsvektor die Anwesenheit der das Fehlen einer anderen eindeutigen Objekt-ID in dem Gruppensatz angibt.



Beschreibung

DATEN VERWANDTER ANMELDUNGEN

[0001] Diese Anmeldung beansprucht die Priorität der US-Patentanmeldung mit der Seriennummer 13/835, 590, die am 15. März 2013 eingereicht wurde und deren Offenbarung hiermit durch Bezugnahme in ihrer Gesamtheit mit eingeschlossen ist.

HINTERGRUND

[0002] Unternehmen speichern häufig Daten in Datenlagern. Derartige Datenlager haben häufig große Volumina an Daten, die über mehrere Datenbanken und Datenbanktabellen verteilt sind. Um daher Daten aus den diversen Tabellen und Datenbanken zusammenzustellen, müssen Datenvereinigungstechniken angewendet werden.

[0003] Tabellen als Aggregat bzw. Vereinigung auf niedriger Ebene (LLA), die eine Zusammenfassung von Daten auf einer niedrigeren Ebene speichern, werden häufig eingesetzt, um das Leistungsvermögen des Vereinigungsprozesses in einem Datenlager zu verbessern. Im Falle einer verteilten Datenbank, die in mehreren Partitionen gespeichert ist, kann eine LLA-Tabelle für jede Partition eingesetzt werden und ein Vereinigungsprozess kann Daten aus jeder der LLA-Tabellen kombinieren, um eine vereinigte Tabelle für die gesamte Datenbank zu erzeugen. Ein derartiger Vereinigungsprozess ist als ein Aufroll-Prozess bekannt und ist nützlich, da eine Zusammenfassung auf höherer Ebene aus einer Zusammenfassung niedrigerer Ebene (die LLA-Tabelle) anstatt von den Ursprungsdaten erzeugt werden kann.

[0004] Aufroll-Prozesse können für eine Aggregat- bzw. Vereinigungsabfrage über mehrere Datenpartitionen im Zusammenhang einer massiv parallelen Verarbeitungs(MPP)Datenbank-Architektur verwendet werden, oder um Abfragen in einer partitionierten Tabelle auszuführen.

[0005] Um Daten in korrekter Weise aus mehreren Tabellen während eines Aufroll-Prozesses zu vereinen, muss die zu Grunde liegende Funktion, die vereinigt wird, additiv sein. Wenn beispielsweise eine erste Tabelle und eine zweite Tabelle jeweils eine Spalte haben, die ausgegebenen Beträgen entsprechen, und ein Anwender wünscht, die Summe aller ausgegebenen Beträge in den beiden Tabellen zu berechnen, kann er additiv die Summe der ausgegebenen Beträge der ersten Tabelle und die Summe der ausgegebenen Beträge der zweiten Tabelle vereinen und die korrekte Gesamtsumme erhalten. Jedoch sind nicht alle Funktionen, die ein Anwender in einem Aufroll-Prozess zu vereinen wünscht, additive Funktionen.

KURZE BESCHREIBUNG DER ZEICHNUNGEN

[0006] Fig. 1A–Fig. 1G zeigen eine Beispieldatenbanktabelle, die Aufzeichnungen von Besuchen von Warenhäusern durch Kunden und einige Beispiele von Vereinigungstabellen auf niedriger Ebene, die aus der Beispieldatenbanktabelle erzeugt sind, enthalten.

[0007] Fig. 2A–Fig. 2B sind Flussdiagramme von Verfahren zur effizienten Ausführung von Operationen an separaten Datenwerten gemäß einer offenbarten Ausführungsform.

[0008] Fig. 3A–Fig. 3C sind Tabellen, die Schritte zur Übersetzung in Zeichen einer oder mehrerer Spalten von Daten in einer Tabelle gemäß einer offenbarten Ausführungsform darstellen.

[0009] Fig. 4A–Fig. 4C zeigen eine LLA-Tabelle, die aus einer in Zeichen übersetzten Tabelle erzeugt ist, und zwei Vereinigungstabellen höherer Ebene als Beispiel, die aus der LLA-Tabelle erzeugt sind.

[0010] Fig. 5A zeigt eine LLA-Tabelle, die aus einer in Zeichen übersetzten Tabelle erzeugt ist, die die Objektabbildungsvektor-Datenstruktur gemäß einer offenbarten Ausführungsform verwendet.

[0011] Fig. 5B zeigt ein Verfahren zur Zuweisung von Werten zu jedem der Bits in dem Objektabbildungsvektor gemäß einer offenbarten Ausführungsform.

[0012] Fig. 5C zeigt eine Tabelle lexikalisch geordneter eindeutiger Datenwerte in einem Bereich gemäß einer offenbarten Ausführungsform.

[0013] Fig. 5D zeigt ein Verfahren zur Zuweisung von Werten zu jedem der Bits in dem Objektabbildungsvektor gemäß einer offenbarten Ausführungsform.

[0014] Fig. 6A zeigt die Schritte, die beim Ausführen eines Aufroll-Prozesses an den Objektabbildungsvektoren beteiligt sind, die in der LLA-Tabelle aus Fig. 5A gespeichert sind, gemäß einer offenbarten Ausführungsform.

[0015] Fig. 6B zeigt eine Tabelle höherer Ebene, die das Ergebnis des Aufroll-Prozesses ist, der in Fig. 6A gezeigt ist.

[0016] Fig. 7A–Fig. 7B zeigen ein Flussdiagramm zur effizienten Ausführung von Operationen an individuellen Datenwerten in einer verteilten Datenbank durch Vereinigen von Bereichen über mehrere Partitionen hinweg gemäß einer offenbarten Ausführungsform.

[0017] Fig. 8A–Fig. 8D zeigen jeweils Schritte, die zum Umwandeln einer Auswahlgruppe an partitionsspezifischen Objekt-IDs bzw. Objekt-Kennungen in globale Objekt-IDs gemäß einer offenbarten Ausführungsform erforderlich sind.

[0018] Fig. 9 zeigt einen Datenwürfel zur Speicherung globaler Objektabbildungsvektoren über mehrere Partitionen hinweg gemäß einer offenbarten Ausführungsform.

[0019] Fig. 10 zeigt einen Aktualisierungsvorgang für ein Datenbanklager unter Anwendung chronologisch zugewiesener Objekt-IDs gemäß einer offenbarten Ausführungsform.

[0020] Fig. 11A zeigt Bereichssynchronisations-Datenstrukturen und einen Prozessablauf gemäß einer offenbarten Ausführungsform.

[0021] Fig. 11B zeigt die Abbildung lokaler Objektabbildungsvektoren auf globale Objektabbildungsvektoren unter Anwendung einer Abbildungstabelle.

[0022] Fig. 12 zeigt eine anschauliche Rechenumgebung, die verwendet werden kann, um das Verfahren zur effizienten Ausführung von Operationen an individuellen Datenwerten gemäß einer offenbarten Ausführungsform auszuführen.

DETAILLIERTE BESCHREIBUNG

[0023] Obwohl Verfahren, Vorrichtungen und computerlesbare Medien hierin durch Beispiele und Ausführungsformen beschrieben sind, erkennt der Fachmann, dass Verfahren, Vorrichtungen und computerlesbare Medien zum effizienten Ausführung von Operationen an individuellen Datenwerten nicht auf die beschriebenen Ausführungsformen oder Zeichnungen beschränkt sind. Es sollte beachtet werden, dass die Zeichnungen und die Beschreibung nicht auf die spezielle offenbarte Form eingeschränkt sein sollen. Vielmehr soll die Erfindung alle Modifizierungen, Äquivalente und Alternativen abdecken, die innerhalb des Grundgedankens und Schutzbereichs der angefügten Patentansprüche liegen. Jegliche Überschriften, die hierin verwendet sind, dienen lediglich dem Zwecke der Strukturierung und sollen den Schutzbereich der Beschreibung oder Ansprüche nicht beschränken. Im hierin verwendeten Sinne wird das Wort „kann“ in einem zulässigen Sinne (d. h., mit der Bedeutung, hat das Potenzial) anstatt in einem bestimmenden Sinne (d. h. Bedeutung muss) verwendet. In ähnlicher Weise bedeuten die Begriffe „einschließen“, „mit“ und „enthält“ einschließend, aber nicht darauf beschränkt.

[0024] Einige Beispiele von Aufroll-Prozessen für additive Funktionen werden nunmehr mit Bezug zu der Tabelle in Fig. 1A beschrieben, die eine beispielhafte Datenbanktabelle zeigt, die Aufzeichnungen von Besuchen in Warenhäusern von Kunden enthält. Die Tabelle enthält Spalten für Kundennamen, Warenhaus-ID bzw. Warenhaus-Kennung und während des Besuchs ausgegebenen Betrag.

[0025] Die Abfragesprache, die durchwegs in dieser Beschreibung verwendet wird, ist die strukturierte Abfragesprache (Structured Query Language, SQL), aber es ist zu beachten, dass eine beliebige Datenbank-Abfragesprache verwendet werden kann, ohne von dem Schutzbereich dieser Erfindung abzuweichen, wozu Kontext bezogene Abfragesprache (CQL), XQuery, YQL, Datalog, OQL, RDQL, multidimensionale Ausdrücke (MDX) und viele andere gehören.

[0026] Eine LLA-Tabelle der Anzahl an Besuchen jedes Warenhauses ist in **Fig. 1B** gezeigt. Die „Anzahl Besuche“-Spalte wird erzeugt, indem die Anzahl an Aufzeichnungen für jedes der Warenhäuser gezählt wird, woraus sich 3 Besuche für Warenhaus 1 und 4 Besuche für Warenhaus 2 ergeben. Wenn der Anwender wünscht, einen Aufroll-Prozess unter Anwendung dieser Tabelle auszuführen, um die Gesamtzahl an Besuchen aller Warenhäuser zu ermitteln, können die Daten mit der Funktion `Sum(Anzahl Besuche)` ermittelt werden, was zu einem Gesamtwert von 7 führt. In diesem Falle muss man nicht zurückgehen zu der ursprünglichen Tabelle, um die Gesamtzahl an Besuchen in Warenhäusern zu ermitteln.

[0027] In ähnlicher Weise ist die Tabelle in **Fig. 1C** eine LLA-Tabelle des Gesamtbetrags, der in jedem der Warenhäuser ausgegeben wurde. Dieser Betrag wird ermittelt, indem die Werte in der „ausgegebener Betrag“-Spalte der ursprünglichen Tabelle für jedes Warenhaus aufsummiert werden. Wenn ein Anwender diese Tabelle aufrollen will, um den Gesamtbetrag zu ermitteln, der in allen Warenhäusern ausgegeben wurde, kann er die Funktionen `Sum(ausgegebener Betrag)` in der LLA-Tabelle aus **Fig. 1C** anwenden, um den korrekten Gesamtbetrag von 107 zu erhalten.

[0028] Die Aggregation bzw. Vereinigung von LLA-Tabellen während eines Aufroll-Prozesses kann auch für Funktionen außerhalb einfacher Summierungen bewerkstelligt werden. **Fig. 1B** zeigt eine LLA-Tabelle des mittleren Betrags, der in jedem Warenhaus pro Besuch ausgegeben wurde. Dieser Betrag wird ermittelt, indem die Werte in der „ausgegebener Betrag“-Spalte der ursprünglichen Tabelle für jedes Warenhaus gemittelt werden. Des Weiteren enthält die LLA-Tabelle eine Zahl für die Anzahl an Besuchen in jedem Warenhaus, wie dies zuvor erläutert ist. Wenn ein Anwender diese Tabelle aufrollen will, um den mittleren Betrag, der in allen Warenhäusern ausgegeben wurde, zu ermitteln, kann er die folgende Funktion verwenden:

$$\text{Durchschnittlich ausgegeben (Gesamt)} = ((\text{Durch}_{\text{Warenhaus1}} \times \text{Besuche}_{\text{Warenhaus1}}) + (\text{Durch}_{\text{Warenhaus2}} \times \text{Besuche}_{\text{Warenhaus2}})) / \text{Summe}(\text{Besuche})$$

[0029] Wiederum erfordert der Aufroll-Prozess nicht die ursprüngliche Tabelle und kann unter Anwendung der LLA-Tabelle alleine bewerkstelligt werden.

[0030] **Fig. 1E** und **Fig. 1F** zeigen LLA-Tabellen, die entsprechend das Minimum, das in jedem Warenhaus ausgegeben wurde, und das Maximum, das in jedem Warenhaus ausgegeben wurde, enthalten. Um die Tabelle in **Fig. 1E** aufzurollen, um einen minimalen ausgegebenen Betrag in einem Warenhaus zu ermitteln, kann der Anwender die Funktion `Min(ausgegebenes Minimum)` auf die LLA-Tabelle anwenden und die korrekte Antwort von 5 erhalten. Um in ähnlicher Weise die Tabelle in **Fig. 1F** aufzurollen, um einen maximal ausgegebenen Betrag in einem beliebigen Warenhaus zu ermitteln, kann der Anwender die Funktion `Max(ausgegebenes Maximum)` auf die LLA-Tabelle anwenden und die korrekte Antwort 31 erhalten.

[0031] Es sei nun auf **Fig. 1G** verwiesen, in der eine LLA-Tabelle gezeigt ist, die eine Zahlenangabe der Anzahl individueller Kunden enthält, die jedes Warenhaus besucht haben. Dies kann mit einer Funktion `Count(distinct<Kundennamen>)` bewerkstelligt werden, die entsprechend der Warenhaus-ID gruppiert wird. In diesem Falle besuchten somit 3 individuelle Kunden (Bill, George und Larry) das Warenhaus 1, und 4 individuelle Kunden (Bill, George, Max und Larry) besuchten Warenhaus 2.

[0032] Wenn ein Anwender wünschte, die LLA-Tabelle in **Fig. 1G** aufzurollen, gäbe es keine Möglichkeit, um in korrekter Weise die Gesamtzahl an individueller Kunden zu bestimmen, die das Warenhaus 1 oder das Warenhaus 2 besucht haben, ohne auf die zu Grunde liegende Haupttabelle in **Fig. 1A** zuzugreifen. Eine `Count-distinct`-Funktion, die auf die Spalte individueller Kunden der LLA-Tabelle angewendet wird, würde lediglich eine Gesamtzahl von 2 individuellen Werten liefern, und eine Summierung würde eine 7 zurückgeben, wobei die korrekte Anzahl individueller Kunden für den Besuch beider Warenhäuser 4 ist. Wenn in ähnlicher Weise die LLA-Tabelle eine Summe aller individuellen Dollar-Beträge, die in jedem Warenhaus ausgegeben wurden, enthielte und der Anwender wünschte, die Summe aus allen individuellen Dollar-Beträge, die in allen Warenhäusern ausgegeben wurden, zu berechnen, müsste er auf die ursprüngliche Tabelle zugreifen, da ansonsten die 11 Dollar, die im Warenhaus 1 ausgegeben wurden, und die 11 Dollar, die im Warenhaus 2 ausgegeben wurden, doppelt gezählt würden. Wenn ferner die LLA-Tabelle Mittelwerte aller individueller Beträge enthielte, die in jedem Warenhaus ausgegeben wurden, wäre ein Aufrollen zur Ermittlung des Mittelwerts aller individueller Beträge, die in beiden Warenhäusern ausgegeben wurden, nicht erfolgreich und es müsste die zu Grunde liegende Haupttabelle abgefragt werden.

[0033] Abfragen, die nicht-additive Befehle beinhalten, etwa `count<distinct>` können Aufroll-Operationen von LLA-Tabelle nicht nutzen, woraus sich ein höherer Berechnungsaufwand ergibt, da diese Befehle mit den zu

Grunde liegenden Daten ausgeführt werden müssen. In einer partitionierten oder anderweitig verteilten Datenbank wird dieser Aufwand vergrößert, da die zu Grunde liegenden Daten vereinigt oder zwischen Partitionen ausgetauscht müssen, um die korrekten Werte abzurufen. Wenn beispielsweise Spalte XYZ über zwei Partitionen hinweg verteilt ist, und wenn der Anwender wünscht, die Anzahl individueller Werte in der Spalte XYZ zu ermitteln, dann müssen alle Werte der Spalte XYZ in der ersten Partition irgendwie mit allen Werten der Spalte XYZ in der zweiten Partition kombiniert werden, um die Anzahl individueller Werte zu ermitteln.

[0034] Die Anmelder haben ein Verfahren und eine zugeordnete Datenstruktur, die als Objektabbildungsvektoren (EMV) bezeichnet ist, entdeckt, die es Anwendern erlaubt, nicht-additive Vereinigungsfunktionen als additiv zu verwenden, wodurch die Ausführung von Aufroll-Prozessen an LLA-Tabellen für Funktionen möglich ist, die zuvor Zugriff auf und die Verarbeitung von zu Grunde liegenden Daten in Tabellen auf niedrigerer Ebene als die LLA-Tabellen erforderten. Die Technik der Anmelder ermöglicht ferner, dass die Datenmenge, die zwischen Knoten einer partitionierten Datenbank ausgetauscht wird, deutlich reduziert ist, da LLA-Tabellen für jeden Knoten selbst für nicht-additive Vereinigungsfunktionen additiv vereinigt werden können.

[0035] Fig. 2A zeigt ein Flussdiagramm eines Verfahrens zur Erzeugung eines EMV gemäß einer offenbaren Ausführungsform. Im Schritt **201** wird eine in Zeichen übersetzte Spalte aus Daten in einer Tabelle einer Datenbank vor der Erzeugung des EMV gespeichert. Dieser Zeichenübersetzungsprozess ist mit Bezug zu den Fig. 3A–Fig. 3C beschrieben. Des Weiteren kann, wie in Fig. 2B gezeigt und später beschrieben ist, ein EMV selbst ohne eine in Zeichen übersetzte Spalte aus Daten auf der Basis einer lexikalischen Ordnung der eindeutigen Datenwerte in einem speziellen Bereich erzeugt werden.

[0036] Fig. 3A zeigt eine Tabelle von Kundentransaktionen vor der Zeichenübersetzung **300**. Die Tabelle enthält Tupel-ID **301** und Datenspalten für Woche **302**, Warenhaus-ID **303**, Kunden-ID **304**, Produkt-ID **305**, Menge **306** und ausgegebener Betrag **307**. Jede der Datenspalten kann einem zugehörigen Bereich entsprechen. Beispielsweise kann ein Bereich, der Kunden-IDs enthält, verwendet werden, um die Kunden-ID-Spalte der gezeigten Tabelle aufzufüllen, aber der gleiche Bereich kann verwendet werden, um eine weitere Tabelle aufzufüllen, die Kunden-IDs mit Kundenadressen auflistet. Selbstverständlich kann ein Bereich auch verwendet werden, um mehrere Spalten in der gleichen Tabelle zu füllen. Beispielsweise kann ein Bereich mit Städtenamen verwendet werden, um Spalten in einer Tabelle zu füllen, die einer Ursprungsstadt und einer Zielstadt entsprechen. So kann die Spalte für die Woche **302** einem Bereich D_Woche entsprechen, die Warenhaus-ID **303** kann einem Bereich D_WarenhausID entsprechen, die Kunden-ID **304** kann einem Bereich D_KundenID entsprechen, Produkt-ID **305** kann einem Bereich D_ProduktID entsprechen, Menge **306** kann einem Bereich D_Qty entsprechen, und ausgegebener Betrag **307** kann einem Bereich D_Ausgegeben entsprechen.

[0037] Um Datenwerte in jeder der Datenspalten in Zeichen zu übersetzen, müssen die eindeutigen Datenwerte, die in jedem der Bereiche (entsprechend den unterschiedlichen Spalten) auftreten, erkannt und auf eindeutige Zeichenwerte abgebildet werden. Abbildungstabelle **310** in Fig. 3B zeigt diese Abbildung eindeutiger Datenwerte auf eindeutige Zeichenwerte für jeden Bereich in der Tabelle. Jeder der eindeutigen Datenwerte in jedem der Bereiche wird auf ein eindeutiges Zeichen für diesen Bereich abgebildet, das als eine Objekt-ID oder EID bezeichnet wird. So enthält beispielsweise D_Woche **312** drei eindeutige Datenwerte und daher werden diese drei eindeutigen Datenwerte auf drei entsprechende eindeutige EIDs abgebildet. In ähnlicher Weise enthält auch D_WarenhausID **313** drei eindeutige Datenwerte und hat drei eindeutige EIDs, D_KundenID **314** hat vier eindeutige Datenwerte und vier EIDs, D_ProduktID hat vier eindeutige Datenwerte und vier EIDs, D_Qty hat fünf eindeutige Datenwerte und fünf EIDs, und C_Ausgegeben hat sechs eindeutige Datenwerte und sechs EIDs.

[0038] Die Abbildungstabellen, die zur Bildung eindeutiger Datenwerte in einem Bereich verwendet werden, so dass sie EIDs entsprechen, können als Nachschlag-Wörterbuch gespeichert werden, so dass Funktionen und zugehörige Prozesse bei Bedarf Datenwerte unter Anwendung von EIDs abrufen können. Des Weiteren ist die Abbildung eindeutiger Datenwerte auf EIDs so gezeigt, dass dies in der lexikalischen Reihenfolge der eindeutigen Datenwerte erfolgt, da dies den Zuweisungsprozess, die künftige Datenspeicherung und Abrufung vereinfachen kann, aber die Zuweisung von EIDs zu eindeutigen Datenwerten kann auch in beliebiger Reihenfolge von eindeutigen Datenwerten erfolgen. Beispielsweise können EIDs zugewiesen werden derart, dass Datenwerte in chronologischer Reihenfolge zugewiesen werden, so dass ein erster Datenwert, der vor einem zweiten Datenwerte angetroffen wird, eine kleinere EID haben wird.

[0039] Fig. 3C zeigt die Tabelle von Kundentransaktionen nach der Übersetzung in Zeichen **320**. Die EIDs aus der Abbildungstabelle **310** haben die Datenwerte in jeder der Spalten aus Daten ersetzt. Jedoch werden in dieser fortgeschrittenen kodierten Struktur jegliche LLA-Tabellen, die über die Tabelle **320** erzeugt sind, Daten

nicht in den geeigneten Format speichern, um Aufroll-Prozesse auszuführen, die nicht-additive (individuell-artige) Vereinigungsoperationen beinhalten.

[0040] Beispielsweise kann eine LLA-Tabelle auf der Grundlage der Transaktionstabelle („Trx“) **320** erzeugt werden, die die Woche, die Warenhaus-ID und die individuellen Kunden, die jedes Warenhaus jede Woche besuchen, unter der folgenden Definition auflisten:

```
Create Table LLA _ TRX as
SelectWoche,
      WarenhausID,
      Count(distinct KundenID) as CD _ KundenID,
from Trx
Group by Woche, WarenhausID;
```

[0041] Diese Definition weist den Wert, der den individuellen Kunden entspricht, die jedes Warenhaus jede Woche besuchten, der Spalte CD_KundenID zu. Des Weiteren erzeugt der „Group by“-Befehl Gruppensätze, die Wochen-Warenhaus-ID-Paaren entsprechen. Beispielsweise Gruppensatz 1 (Woche 1, Warenhaus 1), Gruppensatz 2 (Woche 1, Warenhaus 2), usw. Es sind eine beliebige Anordnung oder Anzahl von Gruppensätzen möglich. Beispielsweise können die Gruppensätze auf einer einzelnen Spalte oder zwei oder mehr Spalten beruhen.

[0042] Gemäß Fig. 4A ist die resultierende LLA_TRX-Tabelle **400** gezeigt. Die Tabelle repräsentiert in genauere Weise die Anzahl individueller Kunden pro Gruppensatz. Beispielsweise listet die Tabelle in korrekter Weise die Anzahl individueller Kunden auf, die Warenhaus 1 in der Woche 1 besuchten, als zwei, oder die Anzahl individueller Kunden, die Warenhaus 1 in Woche 2 besuchten, als drei auf.

[0043] Wenn ein Anwender versuchte, die Tabelle LLA_TRX für jede Woche aufzurollen, um die Anzahl an individuellen Kunden aufzulisten, die beliebige Warenhäuser jede Woche besuchten, wäre er nicht in der Lage, eine korrekte Anzahl für die Anzahl an individuellen Kunden für jeden Gruppensatz erzeugen, wobei die Gruppensätze in diesem Falle Gruppe 1 (Woche 1), Gruppe 2 (Woche 2) usw. sind. Wenn beispielsweise die folgende Definition verwendet wurde, um eine LLA-Tabelle höherer Ebene zu erzeugen:

```
Create Table LLA _ TRX _ WOCHE as
Select Woche,
      Count(distinct CD _ KundenID) as CD _ CD _ KundenID,
from LLA _ Trx
Group by Woche;
```

[0044] Dann würde die resultierende Tabelle, die als Tabelle **410** in Fig. 4B gezeigt ist, nicht in korrekter Weise die Anzahl individueller Kunden auflisten, die entsprechende Warenhäuser jede Woche besuchten. Wenn in ähnlicher Weise eine Summen-Funktion verwendet wurde, um die Tabelle LLA_TRX zu vereinigen:

```
Create Table LLA _ TRX _ WOCHE as
Select Woche,
      Sum(distinct CD _ KundenID) as Sum _ CD _ KundenID,
from LLA _ Trx
Group by Woche;
```

[0045] Dann würde die in Fig. 4C gezeigte resultierende Tabelle ebenfalls in nicht korrekter Weise die Anzahl individueller Kunden, die beliebige Warenhäuser jede Woche besuchten, auflisten. Beispielsweise ist die An-

zahl individueller Kunden in Woche 2 basierend auf der Tabelle **320** in **Fig. 3C** gleich 4 (KundenIDs 1, 2, 3 und 4), aber die Tabelle **410** in **Fig. 4B** listet 2 auf und die Tabelle in **Fig. 4C** **420** listet 5 auf.

[0046] Wie zuvor erläutert ist, liegt dies daran, dass die Vereinigung individueller Werte nicht-additiv ist. Um die korrekten Werte für die Anzahl individueller Kunden pro Woche zu ermitteln, ist es erforderlich, zu der ursprünglichen Datentabelle zurückzukehren, wodurch die LLA_TRX-Tabelle für nicht-additive Vereinigungsabfragen nutzlos wird.

[0047] Die EMV-Datenstruktur, die gemäß einer offenbaren Ausführungsform erzeugt wird, löst dieses Problem, indem die individuellen Werte jedes Gruppensatzes effizient gespeichert werden und in einer Weise, die es erlaubt, dass diese vereinigt werden. Wieder mit Verweis auf **Fig. 2A** kann in einem Schritt **202** eine spezielle Abfrage empfangen werden, die einen oder mehrere Gruppensätze spezifiziert und eine neue Funktion oder einen anderen ähnlichen Befehle enthält, der verwendet wird, um die EMV-Datenstruktur zu erzeugen. Im Schritt **203** kann das Rechensystem einen oder mehrere Objektabbildungsvektoren erzeugen. Selbstverständlich können die Schritte **201**, **202** und **203** in einer anderen Reihenfolge ausgeführt werden. Beispielsweise kann eine Abfrage zuerst empfangen werden (Schritt **202**), dann kann die in Zeichen übersetzte Spalte aus Daten gespeichert werden (Schritt **201**), und dann können ein oder mehrere Objektabbildungsvektoren erzeugt werden (Schritt **202**).

[0048] Um die Schritte **202–203** zu erleichtern, kann eine neue SQL-Funktion `Count(Additive_distinct<Spaltenname>)` definiert werden, um das Rechensystem anzuweisen, einen EMV für jeden Gruppensatz anstelle eines Wertes zu erzeugen. Ein EMV kann ein Bit-Abbildungsvektor sein, wobei jedes Bit auf ein anderes Objekt abgebildet ist. Durch Speicherung der Information, die die individuellen Werte in einem Gruppensatz betrifft, in einem EMV, kann jeder individuelle Wert, der als ein Objekt bzw. eine Entität bezeichnet ist, auf ein einzelnes Bit eines Bit-Abbildungsvektors abgebildet werden. Wenn somit das frühere Beispiel der Zeichen übersetzten Transaktionstabelle (Trx) **320** in **Fig. 3C** verwendet wird, kann die folgende Definition verwendet werden, um die LLA-Tabelle LLA_TRX zu erzeugen:

Create Table LLA _ TRX as

Select Woche,

WarenhausID,

Count(Additive_distinct KundenID) as CD _ KundenID,

from Trx

Group by Woche, WarenhausID;

[0049] **Fig. 5A** zeigt die resultierende LLA_TRX-Tabelle **500**. Wie in der vorhergehenden LLA_TRX-Tabelle ist die Tupel-ID **501**, die jedem Gruppensatz entspricht, Woche **502** und Warenhaus-ID **503** gespeichert. Anstatt der Speicherung des Zählwertes individueller Kunden als ein numerisches Zeichen, werden jedoch die individuellen Kunden für jeden Gruppensatz in einem EMV gespeichert, wobei jedes Bit eine andere EID betrifft und der Wert des Bits die Anwesenheit oder Abwesenheit dieser EID in dem Gruppensatz angibt.

[0050] Selbstverständlich ist eine SQL-Erweiterung, etwa `Additive_distinct`, nicht erforderlich, um das Rechensystem anzuweisen, einen EMV für jeden Gruppensatz zu erzeugen. Beispielsweise kann die Datenbank-Maschine die Erzeugung einer Tabelle mit einer Spalte erfassen, die eine Vereinigungsfunktion individueller Datenwerte beinhaltet, etwa `Count(distinct<Spaltenname>)`, und kann automatisch einen EMV aufbauen, um die Anzahl individueller Datenwerte zu speichern. Dies erlaubt es der EMV-Struktur, in Verbindung mit bestehenden Frontend-Mitteln verwendet zu werden.

[0051] Es wird nun der Aufbau eines EMV mit Bezug zu dem zweiten EMV **513** beschrieben, der dem Gruppensatz (Woche 1, Warenhaus-ID 2) entspricht. Gemäß **Fig. 5B** ist ein leerer EMV **505** gezeigt. Führungszeichen **506** dienen lediglich zur Kennzeichnung des EMV als einen Bit-Abbildungsvektor und können beliebige geeignete Zeichen sein oder können weggelassen werden. Die Längsabmessung des EMV **505** ist die Anzahl an Bit-Stellen und enthält keinerlei Führungszeichen. Die Länge des EMV ist gleich der Gesamtanzahl an EIDs in dem Bereich der Spalte, der zur Erzeugung des EMV verwendet wird. Anders ausgedrückt, die Länge des EMV ist gleich der Anzahl eindeutiger Werte in dem Bereich, der zur Erzeugung des EMV verwendet wird. Wenn somit beispielsweise der Bereich, der zur Erzeugung der EMV für Tabelle **500** in **Fig. 5A** verwendet

wird, der D_KundenID-Bereich ist, dann entnimmt man aus **Fig. 3B**, dass der D_KundenID-Bereich vier EIDs aufweist. Daher wird die Länge der EMVs für diesen Bereich eine Länge von 4 sein mit 4 entsprechenden Bit-Stellen.

[0052] Es sei wieder auf **Fig. 5B** zurückverwiesen, wobei jede der Bit-Stellen einer anderen EID entspricht in der Reihenfolge, in der die EIDs zugewiesen wurden. Somit entspricht die erste Stelle der EID = 1, die zweite Stelle entspricht EID = 2, usw. Um den Wert des Bits jeder der Bit-Stellen in einem EMV zu ermitteln, wird der Gruppensatz, der dem EMV entspricht, geprüft, um zu sehen, ob er den EID-Wert enthält. Somit wird für das erste Bit im EMV **513** ermittelt, ob der Gruppensatz (Woche 1, Warenhaus 2) eine Kunden ID = 1 im Schritt **507** enthält. Aus der Tabelle **320** in **Fig. 3C** erkennen wir, dass eine Kunden ID = 1 in Woche 1 Warenhaus 2 nicht erscheint. Daher ist der Wert des ersten Bits gleich 0.

[0053] In ähnlicher Weise wird für das zweite Bit im EMV **513** ermittelt, ob der Gruppensatz (Woche 1, Warenhaus 2) eine Kunden ID = 2 im Schritt **508** enthält. Da Kunden ID = 2 ebenfalls während der Woche 1 im Warenhaus 2 nicht erscheint, ist der Wert des zweiten Bits ebenfalls null. Für das zweite Bit im EMV **513** wird bestimmt, ob im Schritt **509** der Gruppensatz (Woche 1, Warenhaus 2) eine Kunden ID = 2 enthält. Aus Tabelle **320** erkennt man, dass Kunden ID = 3 zweimal während der Woche 1 Warenhaus 2 erscheint. Daher ist der Wert des zweiten Bits gleich 1. Für das vierte Bit in dem EMV **513** wird ermittelt, ob der Gruppensatz (Woche 1, Warenhaus 2) eine Kunden ID = 4 im Schritt **510** enthält. Da eine Kunden ID = 4 während der Woche 1 im Warenhaus 2 erscheint, wird auch das vierte Bild auf 1 gesetzt. Der resultierende EMV **513** enthält die individuellen Kunden, die Warenhaus 2 während der Woche 1 besuchten, in einem Format auf, das eine additive Vereinigung erlaubt.

[0054] Wenn ein Anwender die Anzahl individueller Kunden pro Gruppensatz bestimmen wollte, muss nur die Summe aller Bits in dem EMV für diesen Gruppensatz bilden. Beispielsweise ist der EMV für den Gruppensatz (Woche 1, Warenhaus 1) gleich OB1100, so dass die Gesamtzahl individueller Kunden $1 + 1 + 0 + 0 = 2$. Der EMV für den Gruppensatz (Woche 2, Warenhaus 1) ist OB1101, so dass die Gesamtzahl individueller Kunden für diesen Gruppensatz $1 + 1 + 0 + 1 = 3$ ist.

[0055] Es sei wieder zurück auf **Fig. 2B** verwiesen wird. Es wird nun ein Verfahren zur Erzeugung von EMVs aus einer nicht in Zeichen übersetzten Datenspalte beschrieben. Schritt **210** ist ähnlich zu dem Schritt **202** aus **Fig. 2A** dahingehend, dass eine Abfrage empfangen wird, die einen oder mehrere Gruppensätze spezifiziert und eine neue Erweiterung oder einen ähnlichen Befehl enthält, der die Datenbank-Maschine anweist, die EMV-Datenstruktur zu erzeugen. Im Schritt **211** wird für jeden der Gruppensätze ein Objektabbildungsvektor auf der Grundlage der lexikalischen Reihenfolge der eindeutigen Datenwerte in dem Bereich erzeugt, der der Spalte aus Daten entspricht. **Fig. 5C** und **Fig. 5D** zeigen diesen Vorgang in größerem Detail im Hinblick auf einen EMV, der für den Gruppensatz (Woche 201001, Warenhaus 1002) in der nicht in Zeichen übersetzten Tabelle **300** in **Fig. 3A** erzeugt ist. Die Tabelle **520** zeigt die eindeutigen Datenwerte in dem Bereich KundenID, der in lexikalischer Reihenfolge angeordnet ist. Der Erzeugungsprozess für den EMV, der in **Fig. 5D** gezeigt ist, verwendet diese lexikalische Reihenfolge, um den Schritt des Erzeugens von EIDs zu umgehen. Wie in dem leeren EMV **525** gezeigt ist, ist die Länge jedes erzeugten EMV gleich der Anzahl an eindeutigen Datenwerten in dem Bereich. Da in diesem Falle vier eindeutige Datenwerte in dem Bereich KundenID vorhanden sind, ist die Länge der EMVs, die für den speziellen Gruppensatz erzeugt sind, vier. Die Position des Bits in jedem EMV entspricht der lexikalischen Position eines entsprechenden eindeutigen Datenwertes in der lexikalisch geordneten Liste aus eindeutigen Datenwerten **520** und der Wert jedes Bits in dem EMV gibt die Anwesenheit oder das Fehlen des entsprechenden eindeutigen Datenwertes in dem Gruppensatz an. Somit wird für das erste Bit in dem EMV **533** ermittelt, ob der Gruppensatz (Woche 201001, Warenhaus 1002) die KundenID enthält, die an der niedrigsten lexikalischen Position liegt. Diese KundenID ist 165, und aus Tabelle **300** in **Fig. 3A** können wir erkennen, dass die KundenID in dem Gruppensatz (Woche 201001, Warenhaus 1002) nicht auftritt, so dass dieses Bit gleich null ist. In ähnlicher Weise bezieht sich das zweite Bit des EMV **533** auf den eindeutigen Datenwert mit der zweitniedrigsten lexikalischen Position, die 167 beträgt, und der somit ebenfalls nicht in dem Gruppensatz auftritt, wodurch sich eine null für das Bit ergibt. Das dritte Bit des EMV **533** betrifft den eindeutigen Datenwert mit der drittniedrigsten lexikalischen Position, die 268 ist, und wir erkennen aus Tabelle **300** in **Fig. 3A**, dass dieser Wert tatsächlich in dem Gruppensatz (Woche 201001, Warenhaus 1002) auftritt, so dass das dritte Bit auf 1 gesetzt wird. In ähnlicher Weise gilt, dass der Wert, der dem vierten Bit **301** entspricht, ebenfalls in dem Gruppensatz auftritt, so dass das vierte Bit auf 1 gesetzt wird. Dies führt zu dem gleichen EMV, der in **Fig. 5B** erzeugt wurde, aber aus einer nicht in Zeichen übersetzten Spalte aus Daten erzeugt werden kann.

[0056] Anders als die vorhergehenden Beispiele an LLA-Tabellen, die den Objektabbildungsvektor nicht benutzen, können die LLA-Tabellen mit EMV in korrekter Weise während eines Aufroll-Prozesses vereinigt werden. Beispielsweise kann die folgende Definition verwendet werden, um eine Tabelle höherer Ebene für die Anzahl individueller Kunden, die ein beliebiges Warenhaus besuchten, aus der LLA-Tabelle, die gemäß Wochen gruppiert ist, zu erzeugen:

```
Create Table LLA _ TRX _ WOCHE as
Select Woche,
      Countdistinct CD _ KundenID) as EMV _ CD _ KundenID,
from LLA _ Trx
Group by Woche;
```

[0057] Gemäß Fig. 6A ist der Vorgang zur Vereinigung der Bit-Vektoren über Warenhäuser und die Gruppierung entsprechend Wochen gezeigt. Jeder der Gruppensätze, der vereinigt werden muss, wird für jede Woche in den Schritten **601A**, **601B** und **601C** ermittelt. Beispielsweise sind die Gruppensätze, die für Woche 1 kombiniert werden müssen, die KundenIDs, die den Gruppensätzen Woche 1, Warenhaus 1 und Woche 1, Warenhaus 2 entsprechen. Die Gruppensätze, die für Woche 2 kombiniert werden müssen, sind die KundenIDs, die den Gruppensätzen Woche 2, Warenhaus 1, Woche 2, Warenhaus 2 und Woche 2, Warenhaus 3 entsprechen. Die Gruppensätze, die für Woche 3 kombiniert werden müssen, sind die Kunden IDs, die den Gruppensätzen Woche 3, Warenhaus 1 und Woche 3, Warenhaus 3 entsprechen.

[0058] Gemäß Schritten **602A**, **602B** und **602C** werden die EMVs für jeden der Gruppensätze, die für jede Woche kombiniert werden müssen, unter Anwendung einer logischen ODER-Verknüpfung vereinigt. Wenn somit beispielsweise zwei EMV mit einer logischen ODER-Verknüpfung kombiniert werden und lediglich der erste EMV eine „1“ in der ersten Position aufweist, dann wird der resultierende EMV eine „1“ an der ersten Position haben.

[0059] Die resultierenden kombinierten EMVs sind in den Schritten **603A**, **603B** und **603C** gezeigt. Jeder EMV entspricht einem anderen Wochen-Gruppensatz. Wie zuvor erläutert ist, kann die Anzahl individueller Kunden in jeder Woche ermittelt werden, indem alle Bits in jedem der EMVs aufsummiert werden.

[0060] Fig. 6B zeigt die resultierende Tabelle **600** mit der Tupel-ID **601** für jeden Gruppensatz, die Wochennummer **602** und die Anzahl individueller Kunden **603** auf der Grundlage der kombinierten EMVs aus Fig. 6A. Durch Verweis auf die ursprüngliche Tabelle **320** in Fig. 3C können wir erkennen, dass die Anzahl individueller Kunden **603** in Tabelle **600** korrekterweise angibt, dass 4 individuelle Kunden (1, 2, 3 und 4) in Woche 1 beliebige Warenhäuser besuchten, 4 individuelle Kunden (1, 2, 3, und 4) in Woche 2 entsprechende Warenhäuser besuchten, und 2 individuelle Kunden (1 und 2) in Woche 3 entsprechende Warenhäuser besuchten.

[0061] Selbstverständlich ist die Vereinigung nicht-additiver Vereinigungsfunktionen nicht auf das Zählen der Anzahl individueller Elemente in dem Gruppensatz beschränkt. Wenn ein Anwender wünscht, einen Satz individueller Werte in einem speziellen Gruppensatz aufzusummieren oder einen Satz individueller Werte in einem Gruppensatz zu mitteln, dann kann dies bewerkstelligt werden, indem die eindeutigen Datenwerte, die jedem Bit des EMV entsprechen, in dem Nachschlag-Wörterbuch abgerufen und diese zu der Summe hinzuaddiert werden, oder diese mit einbezogen werden in den Mittelwert, wenn das Bit „1“ ist. Des Weiteren kann das Nachschlag-Wörterbuch verwendet werden, um jeden der eindeutigen Datenwerte, die den Bits in dem Objektabbildungsvektor entsprechen, für jeden anderen Zweck abzurufen, etwa kundenspezifische Berechnungen oder Maßzahlen. In Situationen, in denen die Objektabbildungsvektoren aus nicht in Zeichen übersetzten Datenwerten auf der Grundlage der lexikalischen Reihenfolge der eindeutigen Datenwerte erzeugt werden, kann die gleiche lexikalische Reihenfolgeabhängigkeit angewendet werden, um die eindeutigen Datenwerte, die jedem Bit entsprechen, abzurufen. Wenn beispielsweise das erste Bit in einem EMV eine „1“ ist, dann kann der eindeutige Datenwert mit der niedrigsten lexikalischen Position in einer Summe, einem Mittelwert oder einer anderen Berechnungen, die Datenwerte in dem Satz betreffen, mit eingeschlossen werden. Wenn in ähnlicher Weise das zweite Bit in einem EMV eine „0“ ist, dann kann der eindeutige Datenwert, der der zweitniedrigsten lexikalischen Position entspricht, aus der Summe, dem Mittelwert oder einer anderen Berechnungen weggelassen werden.

[0062] Es wird nun ein Überblick über eine Abfrageverarbeitung in einer partitionierten Datenbank und einen Prozess zur EMV-basierten Darstellung von Daten in einer partitionierten Datenbank angegeben. Vorausgesetzt sei eine Drei-Partitionen-Datenbank mit einer Transaktionstabelle, wobei eine logische Ansicht dieser Tabelle als eine Ansicht bzw. View repräsentiert werden kann:

Create View V1 as

Select * from Partition1.TRX

Union All

Select * from Partition2.TRX

Union All

Select * from Partition3.TRX

[0063] Die Tabellenpartitionierung ermöglicht eine effiziente verteilte Parallelverarbeitung von Anforderungen mit Ausnahme von nicht-additiven Vereinigungsfunktionen. Wenn beispielsweise eine ursprüngliche Abfrage für die Transaktionstabelle des Namens TRX ähnlich zu der Tabelle **300** in **Fig. 3A** „Select Woche, Sum(Ausgegeben) from TRX group by Woche“ ist, aber die Tabelle über Partitionen verteilt ist, kann die Abfrage in die verteilte Anforderung transformiert werden:

Select Woche, Sum(P _ Ausgegeben) from (

Select Woche, Sum(Ausgegeben) as P _ Ausgegeben from Partition1.TRX group by Woche

Union All

Select Woche, Sum(Ausgegeben) as P _ Ausgegeben from Partition2.TRX group by Woche

Union All

Select Woche, Sum(Ausgegeben) as P _ Ausgegeben from Partition3.TRX group by Woche

)T

Group By Woche

[0064] Für diese verteilte Anforderung kann jedes Abfragesegment, wie „Select Woche, Sum(Ausgegeben) as P_Ausgegeben from Partition1.TRX group by Woche“ unabhängig ausgeführt werden, und ihr Zwischenergebnissatz kann zusammengehängt und schließlich erneut bearbeitet werden, um das endgültige Abfrageergebnis zu erzeugen.

[0065] Im Falle einer nicht-additiven Vereinigungsfunktion, etwa „Select Woche, Count(Distinct KundenID) from TRX group by Woche“ kann die Anforderung in die verteilte Anforderung transformiert werden:

Select Woche, Count(Distinct KundenID) from (

Select Woche, KundenID from Partition1.TRX group by Woche, KundenID

Union All

Select Woche, KundenID from Partition2.TRX group by Woche, KundenID

Union All

Select Woche, KundenID from Partition3.TRX group by Woche, KundenID

)T

Group By Woche

[0066] Die verteilte Anforderung in diesem Falle bewahrt die Liste eindeutiger KundenIDs aus jeder Partition, kann aber den Austausch einer großen Datenmenge zwischen jeder Partition und ein nachfolgendes Verarbeiten dieser Daten erfordern. Beispielsweise muss die Liste an KundenIDs in der ersten Partition für die erste Woche mit der Liste an KundenIDs in der zweiten Partition für die erste Woche verglichen werden, um zu bestimmen, ob es eine Überlappung gibt. Folglich ergibt eine derartige verteilte Anforderung eine Engstelle für den Ausführungsprozess für die Anforderungen.

[0067] Das Verfahren zur Transformation einer nicht-additiven Vereinigungsfunktion in eine additive Funktion kann auf eine oder mehrere partitionierte Datenbanken angewendet werden, um die Datenmenge zu reduzieren, die zwischen Partitionen ausgetauscht wird, und um Leistung zu verbessern. Beispielsweise kann die ursprüngliche Abfrage „Select Woche, Count(Distinct KundenID) from TRX group by Woche“ transformiert werden in:

```

Select Woche, Count(Distinct KundenID) from
  (
    Select Woche, Count(Additive_distinct KundenID) KundenID from Partition1.TRX group by
    Woche
    Union All
    Select Woche, Count(Additive_distinct KundenID) KundenID from Partition2.TRX group by
    Woche
    Union All
    Select Woche, Count(Additive_distinct KundenID) KundenID from Partition3.TRX group by
    Woche
  )T
Group By Woche

```

[0068] Fig. 7A zeigt die Schritte zur Verarbeitung dieser verteilten Anforderung gemäß einer offenbaren Ausführungsform. Die ersten drei Schritte **701** können parallel für jede der Partitionen ausgeführt werden und sind ähnlich zu den Schritten zur Erzeugung eines EMV, der in Fig. 2A gezeigt ist, wobei diese auf jede Partition angewendet werden. Selbstverständlich müssen diese Schritte nicht parallel ausgeführt werden und sie können in beliebiger Reihenfolge ausgeführt werden. Nachdem die ersten drei Schritte **701** für jede der Partitionen ausgeführt sind, ist das Ergebnis drei Sätze an EMVs, einen für jede Partition, wobei jede Stelle des EMV sich auf lokale Partition-EIDs bezieht. Beispielsweise zeigt Fig. 8A drei hypothetische EMV-Sätze **802A**, **802B** und **802C**, die den drei Partitionen entsprechen, zusammen mit drei hypothetischen EID-Abbildungstabellen **801A**, **801B** und **801C**, für jeden der eindeutigen Datenwerte in jeder der Partitionen. Da die EIDs in den Abbildungstabellen **801A–801C** jeweils unterschiedlichen eindeutigen Datenwerten entsprechen, und da die Anzahl an EIDs in jeder Partition unterschiedlich ist, sind die Länge der EMVs und die Signifikanz der Bits in jedem der EMVs in EMV-Sätzen **802A–802C** unterschiedlich. Folglich können die EMV-Sätze **802A–802C** in ihrem aktuellen Zustand nicht durch den Vereinheitlichungsoperator kombiniert werden.

[0069] Es sei wieder auf Fig. 7A verwiesen; im Schritt **702** wird ein Bereichsvereinigungsprozess ausgeführt, um einen globalen Bereich aus jeder der Bereichspartitionen aufzubauen, so dass die EMV-Sätze standardisiert werden können. Weiter mit Bezug zu Fig. 7B werden im Schritt **703** globale EIDs für jeden der eindeutigen Datenwerte in dem globalen Bereich erzeugt. Da die globalen EIDs den gleichen eindeutigen Datenwerten wie die Partition-EIDs entsprechen, kann jede der Partition-EIDs in globale EIDs im Schritt **704** aus Fig. 7B umgewandelt werden. Im Schritt **705** werden die partitionsspezifischen EMVs in globale EMVs umgewandelt, wobei alle die gleiche Länge besitzen und sich auf die gleichen eindeutigen Datenwerte an jedem Bit beziehen. Dies wird bewerkstelligt, indem eine Partition-zu-Globale EID-Abbildungstabelle zur Neubildung jedes EMV verwendet wird, der in der Partition erzeugt ist, um neue zu erzeugen, die den globalen EIDs zugeordnet sind.

[0070] Zur Veranschaulichung dieses Prozesses zeigt Fig. 8B einen globalen Bereich **803**, der sich aus einer Bereichsvereinigung der Bereichspartitionen in Fig. 8A und den entsprechenden globalen EIDs **804** ergibt. Fig. 8C zeigt die Partition-zu-Globale EID-Abbildungstabellen für Partition 1 **811**, Partition 2 **812** und Partition 3 **813**. Wie zuvor erläutert ist, werden die Partition-zu-Globale EID-Abbildungen durch Kreuzreferenzierung eines eindeutigen Datenwerts mit seiner Partition-EID und globaler EID erzeugt. Diese Abbildungstabelle wird dann verwendet, um die partitionsspezifischen EMVs in globale EMVs umzuwandeln. Die resultierenden globalen EMVs, die aus der Umwandlung erzeugt werden, sind in Fig. 8D für die Partition 1 **821**, Partition 2 **822** und Partition 3 **823** gezeigt.

[0071] Sobald die partitionsspezifischen EMVs in globale EMVs umgewandelt worden sind, können sie mit einer logischen ODER-Verknüpfung vereinigt werden, wie dies zuvor erläutert ist. Die resultierenden kombinierten globalen EMVs geben die Anwesenheit oder das Fehlen eindeutiger Datenwerte in einem speziellen

Gruppensatz über alle kombinierte Partitionen hinweg an. Ferner ist die Summe der Bits in einem kombinierten globalen EMV gleich der Gesamtanzahl individueller Datenwerte in dem Gruppensatz, der diesem EMV über Partitionen hinweg entspricht.

[0072] Dieser Bereichsvereinigungsprozess kann für Datenbank-Aktualisierungsoperationen eingesetzt werden. In Datenlager-Umgebungen umfasst der Aktualisierungsprozess häufig das Anhängen neuer Daten. Durch Verarbeitung der angehängten Daten als eine neue Partition zur Erzeugung einer zugehörigen Partition-LLA-Tabelle und durch anschließendes Kombinieren der Partition-LLA-Tabelle mit bestehenden LLA-Tabellen unter Anwendung der zuvor erläuterten Techniken können die neuen Daten mit bestehenden LLA-Tabellen vereinigt werden.

[0073] Beispielsweise können die globalen EMVs für jede der Partitionen in einem Datenwürfel gespeichert werden. **Fig. 9** zeigt einen anschaulichen Datenwürfel **900**, der die globalen EIDs gemäß einer offenbaren Ausführungsform speichert. Wie gezeigt, können die Abmessungen bzw. Dimensionen eine EMV-Abmessung **901**, eine Gruppensatz-Abmessung **902** und eine Partitionsabmessung **903** enthalten. Wenn eine neue Datenaktualisierung empfangen wird, kann diese als eine Partition behandelt werden und kann dem Datenwürfel unter Verwendung des zuvor erläuterten Bereichsvereinigungsprozesses hinzugefügt werden.

[0074] Wie zuvor erläutert ist, können EIDs auch in chronologischer Reihenfolge des Auftretens der Datenwerte anstatt in lexikalische Reihenfolge der Datenwerte zugewiesen werden. Durch die Zuweisung von EIDs auf diese Weise kann der Bereichsvereinigungsschritt für eine Aktualisierung für ein Datenlager vermieden werden. Gemäß **Fig. 10** ist ein bestehendes Datenlager bei **1001** gezeigt. In diesem Beispiel wird die chronologische Reihenfolge, in der die Datenwerte verarbeitet werden, angewendet, um die EIDs für jeden Datenwert zuzuweisen. Die Tabelle **1002** zeigt die EIDs für vier Datenwerte, die in der folgenden Reihenfolge verarbeitet werden: 368, 167, 165, 268. Des Weiteren zeigt die Tabelle **1003** die EMV für drei Gruppensätze, die Datenwerte enthalten. Somit entspricht beispielsweise der zweite EMV „081110“ einem Gruppensatz, der die Datenwerte 368, 167 und 165 enthält.

[0075] **Fig. 10** zeigt ebenfalls eine Datenaktualisierung **1004** für das Datenlager. Die Datenaktualisierung ist ähnlich zu den früher erläuterten Partitionen. Eine lokale EID-Abbildungstabelle **1005** zeigt die EIDs für die Datenwerte in der Aktualisierung, die ebenfalls in chronologischer Reihenfolge zugewiesen sind. Des Weiteren sind die EMVs für zwei Gruppensätze bei **1006** gezeigt. In diesem Falle sind zwei der Datenwerte in der Aktualisierung neu, wobei einer der Datenwerte bereits in dem Datenlager existiert.

[0076] Das aktualisierte Datenlage ist bei **1007** gezeigt. Da die EIDs in chronologischer Reihenfolge zugewiesen sind, muss ein Bereichsvereinigungsprozess nicht ausgeführt werden. Vielmehr werden die zwei neuen Datenwerte in der Datenaktualisierung, 392 und 163, den nächsten beiden verfügbaren EIDs, 5 und 6 in diesem Falle, zugewiesen. Da es keine erneute Abbildung auf bestehende EIDs gibt, kann die bestehende Tabelle aus EMV **1003** einfach dadurch aktualisiert werden, dass die Länge jedes EMV erweitert wird, indem eine Anzahl an Nullen gleich der Anzahl neuer EIDs am Ende jedes EMV hinzugefügt wird. Dies ergibt eine Tabelle **1009** mit den aktualisierten EMVs. Selbstverständlich müssen die Nullen nicht notwendigerweise den EMVs hinzugefügt werden, da die Abwesenheit eines Zeichens an einer speziellen Position als eine Null interpretiert werden kann.

[0077] Die einzige Neuabbildung, die für die Datenaktualisierung erforderlich ist, wenn die EIDs auf der Grundlage einer chronologischen Reihenfolge der Datenwerte zugewiesen werden, erfolgt auf die EMVs in der tatsächlichen Datenaktualisierung. In diesem Falle müssen die EMVs in der Aktualisierung **1006** erneut abgebildet werden, um die globale EID-Tabelle **1008** anstelle der lokalen EID-Tabelle **1005** zu verwenden, um die aktualisierten EMVs in der Tabelle **1010** zu erzeugen. Da somit beispielsweise der erste EMV in der Aktualisierung **1006** einem Gruppensatz entspricht, der die Werte 392 und 163 enthält (1 an der ersten und der dritten Position des EMV), und die globalen EIDs, die diesen Datenwerten entsprechen, 5 und 6 sind, hat der erste EMV in der aktualisierten Datentabelle **1010**, die dem ersten EMV in der Datenaktualisierung entspricht, eine 1 an der fünften und der sechsten Position.

[0078] In dem Szenario, in welchem ein oder mehrere partitionsspezifische EMVs aus einer nicht in Zeichen übersetzten Spalte aus Daten in einer ersten Datenpartition erzeugt werden und eine Vereinigung mit einer oder mehreren anderen Partitionen ausgeführt wird, kann die erste Bereichspartition, die der nicht in Zeichen übersetzten Spalte aus Daten in der ersten Datenpartition entspricht, mit einer oder mehreren anderen Bereichspartitionen in dem Satz aus Bereichspartitionen kombiniert werden, um einen globalen Bereich zu erzeugen. Dieser globale Bereich kann dann verwendet werden, um einen oder mehrere globale EMVs, die dem

einen oder den mehreren partitionensspezifischen EMVs entsprechen, auf der Grundlage der lexikalischen Reihenfolge aller eindeutigen Datenwerte in dem globalen Bereich zu erzeugen, ähnlich zu dem Prozess der Erzeugung eines EMV, der mit Bezug zu **Fig. 2B** beschrieben ist. Die Länge jedes globalen EMV kann gleich der Anzahl eindeutiger Datenwerte in dem globalen Bereich sein, wobei die Position jedes Bits in einem globalen EMV der lexikalischen Position eines entsprechenden eindeutigen Datenwertes in einer lexikalisch geordneten Liste der eindeutigen Datenwerte in dem globalen Bereich entsprechen kann, und der Wert jedes Bits in einem globalen EMV kann die Anwesenheit oder das Fehlen des entsprechenden eindeutigen Datenwertes in einem Gruppensatz angeben.

[0079] Es wird nun ein Bereichssynchronisierungsprozess beschrieben, der angewendet werden kann, um den Bereichsvereinigungsprozess, der zuvor beschrieben ist, zu umgehen, und der das Betreiben einer Datenbank in einem verteilten System effizienter machen kann. **Fig. 11A** zeigt zwei Partitionen, **1101** und **1102**, in einer verteilten Datenbank. Ähnlich zu der verteilten Datenbank aus **Fig. 8A** enthält jede der Partitionen eine lokale EID-Tabelle, wobei Tabelle **1104** für die Partition **1101** und Tabelle **1106** für Partition **1102** dient. In diesem Beispiel sind die EIDs so gezeigt, dass sie auf der Grundlage einer lexikalischen Reihenfolge der Datenwerte zugewiesen sind, aber die EIDs können auch chronologisch zugewiesen werden, wie dies zuvor erläutert ist. Des Weiteren enthält die verteilte Datenbank eine globale EID-Tabelle **1103**. Die globale EID-Tabelle **1103** enthält Abbildungen aller Datenwerte auf globale EIDs. Wiederum können diese globalen EIDs auf der Grundlage lexikalischer Reihenfolge von Datenwerten zugewiesen, aber sie können auch auf der Grundlage einer chronologischen Reihenfolge von Datenwerten zugewiesen werden. Obwohl die globale EID-Tabelle **1103** so gezeigt ist, dass sie außerhalb der Partition **1101** und der Partition **1102** gespeichert ist, kann die globale EID-Tabelle auch in einer oder beiden Partitionen gespeichert werden.

[0080] Jede der Partitionen **1101** und **1102** enthält ferner eine lokale EID-zu-globale EID-Abbildungstabelle, d. h. Tabelle **1105** für Partition **1101** und Tabelle **1107** für Partition **1102**. Diese Abbildungstabellen können verwendet werden, um lokale EMVs in globale EMVs zu übersetzen, wie dies später unter Anwendung der Beispiele der EMVs **1108** und **1109** erläutert ist.

[0081] Wenn eine Aktualisierung in der Partition **1101** oder in der Partition **1102** empfangen wird, können jegliche neuen eindeutigen Datenwerte zu der globalen EID-Tabelle **1103** gesendet werden. Diese neuen eindeutigen Datenwerte können verwendet werden, um die globale EID-Tabelle zu aktualisieren. Wenn beispielsweise die EIDs durch lexikalische Reihenfolge der eindeutigen Datenwerte zugewiesen werden, können die Datenwerte in der globalen EID-Tabelle **1103** neu sortiert werden und die globalen EIDs können erneut jedem eindeutigen Datenwerte zugewiesen werden, wobei die neun empfangenen eindeutigen Datenwerte berücksichtigt werden. Zu beachten ist, dass, wenn chronologisch zugewiesene EIDs verwendet werden, dieser Vorgang stark vereinfacht wird, da neue eindeutige Datenwerte der nächsten verfügbaren EID zugewiesen werden, und es liegt kein Bedarf vor, EIDs erneut allen vorher existierenden eindeutigen Datenwerten zuzuordnen.

[0082] Nachdem die globale EID-Tabelle **1103** aktualisiert ist, können jegliche neue globale EIDs zurück zu den Partitionen **1101** und **1102** gesendet werden. Die neuen globalen EIDs können verwendet werden, um jede lokale EID-zu-Globale EID-Abbildungstabelle **1105** und **1107** zu aktualisieren. Wiederum sind chronologisch zugewiesene EIDs für diesen Vorgang effizienter, da lediglich neue globale EIDs für die neuen eindeutigen Datenwerte vorgesehen sind, wohingegen bei lexikalisch zugewiesenen EIDs alle EIDs für bestehende eindeutige Datenwerte möglicherweise neu konfiguriert werden müssen.

[0083] Wenn eine Abfrage EMVs aus mehr als einer Partition erfordert, können die lokalen EID-zu-globale EID-Abbildungstabellen **1105** und **1107** verwendet werden, um jegliche lokale EMVs in globale EMVs umzuwandeln. Gemäß **Fig. 11B** ist der Umwandlungsprozess für einen beispielhaften EMV **1108** in der Partition **1101** gezeigt. Wie in **Fig. 11B** gezeigt ist, wird jedes Bit in dem lokalen EMV in seiner lokalen Position auf seine globale Position in dem äquivalenten globalen EMV **1110** abgebildet. Da somit beispielsweise die lokale EID-zu-globale EID-Abbildungstabelle **1105** die lokale EID „3“ auf die globale EID „5“ abbildet, wird der Wert des Bits an der zweiten Position in dem lokalen EMV **1108** der fünften Position des globalen EMV **1110** zugewiesen. Wenn es keine Abbildung einer Position in dem globalen EMV gibt, etwa für die vierte Position in dem globalen EMV **1110**, dann kann dieser Wert auf null gesetzt werden, um das Fehlen des speziellen eindeutigen Datenwertes in dem Gruppensatz wiederzugeben, da, wenn dieser Wert vorhanden war, er in der lokalen EID-zu-globale EID-Abbildungstabelle **1105** wäre. In ähnlicher Weise kann die lokale EID-zu-globale EID-Abbildungstabelle **1107** verwendet werden, um den globalen EMV **1111** aus dem lokalen EMV **1109** in der Partition **1102** zu erzeugen.

[0084] Mittels dieses Bereichssynchronisierprozesses kann der Bereichvereinigungsvorgang, der zuvor beschrieben ist, umgangen werden, woraus sich eine Verbesserung der Verarbeitungszeit ergibt und die Anzahl eindeutiger Datenwerte verringert wird, die in einer gegebenen Zeit exportiert werden müssen. Selbstverständlich ist ein einzelner EMV in jeder Partition nur zum Zwecke der Klarheit gezeigt. Wenn in der Praxis eine lokale EID und ein Wert in einer speziellen Partition vorhanden sind, dann müssten ein gewisser Gruppensatz und ein EMV diesen Wert enthalten. Des Weiteren sind zwei Partitionen nur zum Zwecke der Klarheit gezeigt, und die Datenbank kann eine beliebige Anzahl von Partitionen enthalten.

[0085] Eine oder mehrere der zuvor beschriebenen Techniken können implementiert werden in einem oder mehreren Computersystemen oder darin enthalten sein. **Fig. 12** zeigt ein verallgemeinertes Beispiel einer Rechenumgebung **1200**. Die Rechenumgebung **1200** beabsichtigt nicht, eine Beschränkung im Hinblick auf den Bereich der Verwendung oder Funktion einer beschriebenen Ausführungsformen aufzuerlegen.

[0086] Mit Bezug zu **Fig. 12** umfasst die Rechenumgebung **1200** mindestens eine Verarbeitungseinheit **1210** und einen Speicher **1220**. Die Verarbeitungseinheit **1210** führt computerausführbare Befehle aus und kann ein realer oder virtueller Prozessor sein. In einem Mehr-Verarbeitungssystem führen mehrere Verarbeitungseinheiten computerausführbare Befehle aus, um die Verarbeitungsleistung zu erhöhen. Der Speicher **1220** kann ein flüchtiger Speicher (beispielsweise Register, Cache-Speicher, RAM), ein nicht-flüchtiger Speicher (beispielsweise ROM, EEPROM, ein Flash-Speicher, usw.) oder eine Kombination dieser beiden Arten sein. Der Speicher **1220** kann Software **1280**, die die oben beschriebenen Techniken implementiert, speichern.

[0087] Eine Rechenumgebung kann weitere Merkmale aufweisen. Beispielsweise kann die Rechenumgebung **1200** einen Speicher **1240**, ein oder mehrere Eingabegeräte **1250**, ein oder mehrere Ausgabegeräte **1260** und eine oder mehrere Kommunikationsverbindungen **1290** aufweisen. Ein Zwischenverbindungsmechanismus **1270**, etwa ein Bus, eine Steuerung, oder ein Netzwerk verbinden die Komponenten der Rechenumgebung **1200**. Typischerweise stellt eine Software eines Betriebssystems oder eine Firmware (nicht gezeigt) eine Betriebsumgebung für andere Software bereit, die in der Rechenumgebung **1200** ausgeführt wird, und diese koordiniert Aktivitäten der Komponenten der Rechenumgebung **1200**.

[0088] Der Speicher **1240** kann entfernbar oder stationär sein, und umfasst magnetische Disketten, Magnetbänder oder Kassetten, CD-ROM, CD-RW, DVD, oder ein anderes Medium, das zur Speicherung von Information verwendet werden kann und auf das in der Rechenumgebung **1200** zugegriffen werden kann. Der Speicher **1240** kann Befehle für die Software **1280** speichern.

[0089] Das bzw. die Eingabegerät bzw. Geräte **1250** können ein berührungsempfindliches Eingabegerät, etwa eine Tastatur, eine Maus, ein Stift, ein Eingabeball, berührungsempfindliche Bildschirme oder eine Spielesteuerung, ein Spracheingabegerät, ein Abtastgerät, eine digitale Kamera, eine Fernsteuerung, oder ein anderes Gerät sein, das Eingabe an die Rechenumgebung **1200** liefert. Das bzw. die Ausgabegerät bzw. Geräte **1260** können eine Anzeige, ein Fernseher, ein Monitor, ein Drucker, Lautsprecher oder ein anderes Gerät sein, das Ausgabe aus der Rechenumgebung **1200** bereitstellt.

[0090] Die Kommunikationsverbindung(en) **1290** ermöglicht(en) eine Verbindung über ein Kommunikationsmedium zu einer weiteren Recheneinheit. Das Kommunikationsmedium übermittelt Information, etwa computerausführbare Befehle, Audio- oder Video-Information oder andere Daten in einem modulierten Datensignal. Ein moduliertes Datensignal ist ein Signal, das eine oder mehrere seiner Eigenschaften so festgelegt oder geändert hat, dass Information auf das Signal kodiert ist. Beispielsweise, ohne einschränkend zu sein, kann das Kommunikationsmedium drahtgestützte oder drahtlose Techniken beinhalten, die mit einem elektrischen, optischen, HF, infraroten, akustischen oder anderen Trägern implementiert sind.

[0091] Implementierungen können im allgemeinen Zusammenhang von computerlesbaren Medien beschrieben werden. Computerlesbare Medien sind beliebige verfügbare Medien, auf die in einer Rechenumgebung zugegriffen werden kann. Beispielsweise, ohne einschränkend zu sein, umfassen in der Rechenumgebung **1200** computerlesbare Medien den Speicher **1220**, den Speicher **1240**, Kommunikationsmedien und Kombinationen der zuvor genannten Medien.

[0092] Selbstverständlich zeigt **Fig. 12** die Rechenumgebung **1200**, das Anzeigegerät **1260** und das Eingabegerät **1250** als individuelle Einheiten, um diese einfach erkennen zu können. Die Rechenumgebung **1200**, das Anzeigegerät **1260** und das Eingabegerät **1250** können separate Geräte (beispielsweise ein Personalcomputer, der über Kabel mit einem Monitor und einer Maus verbunden ist) sein, können in einer einzigen Einrichtung (beispielsweise ein Mobilgerät mit einer berührungsempfindlichen Anzeige, etwa ein intelligentes Telefon

oder ein Tablett) sein, oder können Kombinationen von Geräten sein (beispielsweise ein Rechengerät, das funktionsmäßig mit einer berührungsempfindlichen Bildschirmanzeige verbunden ist, mehrere Recheneinrichtungen, die mit einem einzigen Anzeigegerät und Eingabegerät verbunden sind, usw.). Die Rechenumgebung **1200** kann eine digitale Fernseh-Box, ein Personalcomputer oder ein oder mehrere Dienstleister, beispielsweise eine Ansammlung von vernetzten Dienstleister-Rechnern, eine Cluster-Dienstleister-Umgebung, oder ein Cloud- bzw. Wolke-Netzwerk von Recheneinrichtungen sein.

[0093] Nach Beschreibung und Darstellung der Prinzipien unserer Erfindung mit Bezug zu der beschriebenen Ausführungsform erkennt man, dass die beschriebene Ausführungsform in Aufbau und Detail modifiziert werden kann, ohne von derartigen Prinzipien abzuweichen. Es sollte beachtet werden, dass die Programme, Prozesse oder Verfahren, die hierin beschrieben sind, nicht auf die spezielle Art einer Rechenumgebung bezogen oder darauf beschränkt sind, sofern dies nicht anders angegeben ist. Diverse Arten von Rechenumgebungen für Allgemeinzwecke oder spezialisierte Rechenumgebungen können in Verbindung mit den hierin beschriebenen Lehren verwendet werden oder Operationen in Übereinstimmung mit diesen Lehren ausführen. Elemente der beschriebenen Ausführungsformen, die in Software gezeigt sind, können in Hardware eingerichtet werden, und umgekehrt.

[0094] Im Hinblick auf die vielen möglichen Ausführungsformen, auf die die Prinzipien unserer Erfindung anwendbar sind, beanspruchen wir als unsere Erfindung alle derartigen Ausführungsformen, die innerhalb des Schutzbereichs und Grundgedankens der folgenden Patentansprüche und ihrer Äquivalente liegen.

Patentansprüche

1. Ein Verfahren zur effizienten Ausführung von Operationen an individuellen Datenwerten durch eine oder mehrere Recheneinrichtungen, wobei das Verfahren umfasst:

Speichern, durch mindestens eine der einen oder mehreren Recheneinrichtungen, eine in Zeichen übersetzte Spalte aus Daten in einer Tabelle, wobei die in Zeichen übersetzte Spalte aus Daten erzeugt wird, durch Abbilden jedes eindeutigen Datenwertes in einem Bereich, der einer Spalte aus Daten entspricht, auf eine Objekt-ID, und durch Ersetzen jedes Datenwertes in der Spalte durch eine entsprechende Objekt-ID, um die Spalte aus Zeichen übersetzter Daten, die eine oder mehrere Objekt-IDs enthält, zu erzeugen;

Empfangen, durch mindestens eine der einen oder mehreren Recheneinrichtungen, einer Abfrage, die die Spalte aus Daten betrifft, wobei die Abfrage einen oder mehrere Gruppensätze zur Gruppierung der Daten definiert, die in Reaktion auf die Abfrage abgerufen werden; und

Erzeugen, durch mindestens eine der einen oder mehreren Recheneinrichtungen, eines Objektabbildungsvektors für jeden Gruppensatz in dem einen oder den mehreren Gruppensätzen, wobei die Länge jedes Objektabbildungsvektors gleich der Gesamtanzahl an Objekt-IDs für den Bereich ist, und wobei der Wert jedes Bits in jedem Objektabbildungsvektor die Anwesenheit oder das Fehlen einer anderen Objekt-ID in dem entsprechenden Gruppensatz angibt.

2. Das Verfahren nach Anspruch 1, wobei die Abbildungen zwischen jeder der Objekt-IDs und den eindeutigen Datenwerten in einem Nachschlag-Wörterbuch gespeichert werden.

3. Das Verfahren nach Anspruch 1, wobei ein Wert von „1“ für ein Bit in einem Objektabbildungsvektor die Anwesenheit einer Objekt-ID in dem Gruppensatz angibt, und ein Wert „0“ für ein Bit in einem Objektabbildungsvektor das Fehlen der Objekt-ID in dem Gruppensatz angibt.

4. Das Verfahren nach Anspruch 1, wobei die Anzahl individueller Datenwerte in einem beliebigen Gruppensatz gleich der Summe der Werte jedes Bits des Objektabbildungsvektors ist, der diesem Gruppensatz entspricht.

5. Das Verfahren nach Anspruch 1, wobei der eine oder die mehreren Gruppensätze mindestens zwei Gruppensätze umfassen, wobei das Verfahren ferner umfasst:

Empfangen, durch mindestens eine der einen oder mehreren Recheneinrichtungen, eines Befehls zur Vereinigung der Daten aus zwei oder mehr der Gruppensätze in einen kombinierten Gruppensatz; und

Ausführen, durch mindestens eine der einen oder mehreren Recheneinrichtungen, einer logischen ODER-Operation an den Objektabbildungsvektoren, die den zwei oder mehr Gruppensätzen entsprechen, um einen kombinierten Objektabbildungsvektor zu erzeugen.

6. Das Verfahren nach Anspruch 5, wobei die Anzahl individueller Datenwerte in dem kombinierten Gruppensatz gleich der Summe der Werte jedes der Bits des kombinierten Objektabbildungsvektors ist.

7. Das Verfahren nach Anspruch 1, wobei die Tabelle partitioniert und über mehrere Rechenknoten verteilt wird, und wobei die Spalte aus Daten eine erste Spaltenpartition in einem Satz aus Spaltenpartitionen ist, wobei der Bereich eine erste Bereichspartition in einem Satz aus Bereichspartitionen ist, wobei die eine oder die mehreren Objekt-IDs eine oder mehrere erste Partitionsobjekt-IDs und die Objektabbildungsvektoren ein oder mehrere erste Partitionsobjektabbildungsvektoren sind, wobei das Verfahren ferner umfasst:

Kombinieren, durch mindestens eine der einen oder mehreren Recheneinrichtungen, der ersten Bereichspartition mit einer oder mehreren anderen Bereichspartitionen in dem Satz aus Bereichspartitionen, um einen globalen Bereich zu erzeugen;

Übersetzen in Zeichen, durch mindestens eine der einen oder mehreren Recheneinrichtungen, eines oder mehrerer der eindeutigen Datenwerte in dem globalen Bereich, indem jeder des einen oder mehreren eindeutigen Datenwerte auf eine globale Objekt-ID abgebildet wird, um eine oder mehrere globale Objekt-IDs zu erzeugen;

Abbilden, durch mindestens eine der einen oder mehreren Recheneinrichtungen, jede der einen oder mehreren ersten Partitionsobjekt-IDs auf eine entsprechende globale Objekt-ID in der einen oder den mehreren globalen Objekt-IDs durch Kreuzreferenzierung eines gemeinsamen eindeutigen Datenwertes; und

Übersetzen, durch mindestens eine der einen oder mehreren Recheneinrichtungen, jedes des einen oder der mehreren ersten Partitionsobjektabbildungsvektoren in einen oder mehrere globale Objektabbildungsvektoren auf der Grundlage der Abbildung einer oder mehrerer erster Partitionsobjekt-IDs auf eine oder mehrere globale Objekt-IDs.

8. Das Verfahren nach Anspruch 7, das ferner umfasst:

Ausführen, durch mindestens eine der einen oder mehreren Recheneinrichtungen, einer logischen ODER-Operation an dem einen oder den mehreren globalen Objektabbildungsvektoren, die der ersten Partition entsprechen, und an einem oder mehreren zweiten globalen Objektabbildungsvektoren, die einer zweiten Partition entsprechen, um einen oder mehrere kombinierte globale Objektabbildungsvektoren zu erzeugen.

9. Das Verfahren nach Anspruch 8, wobei die Summe der Werte jedes Bits jedes der kombinierten globalen Objektabbildungsvektoren für die erste Partition und die zweite Partition die Gesamtzahl individueller Elemente in einem speziellen Gruppensatz ist, der diesen globalen Objektabbildungsvektoren entspricht.

10. Das Verfahren nach Anspruch 7, wobei die globalen Objektabbildungsvektoren für jede Partition in einem globalen Datenwürfel gespeichert werden, wobei der globale Datenwürfel eine Objektabbildungsvektorenabmessung, eine Gruppensatzabmessung und eine Partitionsabmessung aufweist.

11. Das Verfahren nach Anspruch 1, wobei die Anzahl an Gruppensätzen zumindest teilweise durch die Anzahl individueller Datenwerte in einer zweiten Spalte aus Daten in der Tabelle bestimmt ist.

12. Das Verfahren nach Anspruch 1, wobei die Gruppensätze in Bezug auf eine oder mehrere zweite Spalten aus Daten in der Tabelle definiert werden.

13. Das Verfahren nach Anspruch 1, wobei jede Objekt-ID auf jeden eindeutigen Datenwert in chronologischer Reihenfolge des Auftretens der eindeutigen Datenwerte abgebildet wird, so dass die erste Objekt-ID auf den ersten eindeutigen angetroffenen Datenwert abgebildet wird.

14. Das Verfahren nach Anspruch 13, wobei die eine oder die mehreren Objekt-IDs eine oder mehrere globale Objekt-IDs sind und der eine oder die mehreren Objektabbildungsvektoren einer oder mehrere vorher existierende globale Objektabbildungsvektoren sind, wobei das Verfahren ferner umfasst:

Empfangen, durch mindestens eine der einen oder mehreren Recheneinrichtungen, einer Aktualisierung mit einem oder mehreren neuen Objektabbildungsvektoren, die einer oder mehreren neuen Objekt-IDs entsprechen, wobei jede neue Objekt-ID einem eindeutigen Datenwert entspricht;

Zuweisen, durch mindestens eine der einen oder mehreren Recheneinrichtungen, einer neuen globalen Objekt-ID zu jedem eindeutigen Datenwert, der nicht bereits in dem Bereich repräsentiert ist; und

Erzeugen, durch mindestens eine der einen oder mehreren Recheneinrichtungen, eines oder mehrerer globaler Objektabbildungsvektoren, die dem einen oder den mehreren neuen Objektabbildungsvektoren entsprechen, wobei die Länge jedes globalen Objektabbildungsvektors gleich der Gesamtzahl globaler Objekt-IDs ist, und wobei der Wert jedes Bits in jedem globalen Objektabbildungsvektor die Anwesenheit oder das Fehlen einer anderen globalen Objekt-ID in einem entsprechenden Gruppensatz angibt.

15. Das Verfahren nach Anspruch 14, das ferner umfasst:

Aktualisieren, durch mindestens eine der einen oder mehreren Recheneinrichtungen, der vorher existierenden globalen Objektabbildungsvektoren durch Anhängen einer Anzahl an Nullen an das Ende jedes vorher existierenden globalen Objektabbildungsvektors, wobei die Anzahl an Nullen gleich der Anzahl eindeutiger Datenwerte in der Aktualisierung ist, die nicht bereits in dem Bereich repräsentiert waren.

16. Das Verfahren nach Anspruch 1, wobei jede Objekt-ID auf jeden eindeutigen Datenwert in lexikalischer Reihenfolge der eindeutigen Datenwerte abgebildet wird, so dass die erste Objekt-ID auf den eindeutigen Datenwert abgebildet wird, der als erstes in der lexikalischen Reihenfolge der eindeutigen Datenwerte auftritt.

17. Das Verfahren nach Anspruch 1, wobei die numerische Platzierung jedes Bits in einem Objektabbildungsvektor der Objekt-ID entspricht, deren Anwesenheit oder Fehlen in dem Gruppensatz das Bit repräsentiert.

18. Das Verfahren nach Anspruch 1, wobei Speichern einer in Zeichen übersetzten Spalte aus Daten nach dem Empfang einer Abfrage erfolgt.

19. Das Verfahren nach Anspruch 1, wobei Speichern einer in Zeichen übersetzten Spalte aus Daten vor dem Empfang einer Abfrage erfolgt.

20. Das Verfahren nach Anspruch 1, wobei die Abfrage einen Summenbefehl enthält, der jeden des einen oder der mehreren Gruppensätze betrifft, und wobei das Verfahren ferner umfasst:
für jeden der einen oder mehreren Gruppensätze Summieren, durch mindestens eine der einen oder mehreren Recheneinrichtungen, der eindeutigen Datenwerte, die den Objekt-IDs entsprechen, die als anwesend in dem Objektabbildungsvektor, der diesem Gruppensatz entspricht, angegeben sind.

21. Das Verfahren nach Anspruch 1, wobei die Abfrage einen Mittelwertbefehl enthält, der sich auf jeden des einen oder der mehreren Gruppensätze bezieht, und wobei das Verfahren ferner umfasst:
für jeden des einen oder der mehreren Gruppensätze, Berechnen, durch mindestens eine der einen oder mehreren Recheneinrichtungen, des Mittelwerts der eindeutigen Datenwerte, die den Objekt-IDs entsprechen, die als anwesend in dem Objektabbildungsvektor, der diesem Gruppensatz entspricht, angegeben sind.

22. Eine Vorrichtung zur effizienten Ausführung von Operationen an individuellen Datenwerten, wobei die Vorrichtung umfasst:

einen oder mehrere Prozessoren; und

einen oder mehrere Speicher, die funktionsmäßig mit mindestens einem der einen oder mehreren Prozessoren verbunden sind und darin gespeicherte Befehle aufweisen, die, wenn sie von mindestens einem der einen oder mehreren Prozessoren ausgeführt werden, mindestens einen der einen oder mehreren Prozessoren veranlassen, um:

eine in Zeichen übersetzte Spalte aus Daten in einer Tabelle zu speichern, wobei die in Zeichen übersetzte Spalte aus Daten erzeugt wird, durch Abbilden jedes eindeutigen Datenwertes in einem Bereich, der einer Spalte aus Daten entspricht, auf eine Objekt-ID, und durch Ersetzen jedes Datenwertes in der Spalte durch eine entsprechende Objekt-ID, um die Spalte aus Zeichen übersetzter Daten, die eine oder mehrere Objekt-IDs enthält, zu erzeugen;

eine Abfrage zu empfangen, die die Spalte aus Daten betrifft, wobei die Abfrage einen oder mehrere Gruppensätze zur Gruppierung der Daten definiert, die in Reaktion auf die Abfrage abgerufen werden; und

einen Objektabbildungsvektor für jeden Gruppensatz in dem einen oder den mehreren Gruppensätzen zu erzeugen, wobei die Länge jedes Objektabbildungsvektors gleich der Gesamtanzahl an Objekt-IDs für den Bereich ist, und wobei der Wert jedes Bits in jedem Objektabbildungsvektor die Anwesenheit oder das Fehlen einer anderen Objekt-ID in dem entsprechenden Gruppensatz angibt.

23. Die Vorrichtung nach Anspruch 22, wobei die Abbildungen zwischen jeder der Objekt-IDs und den eindeutigen Datenwerten in einen Nachschlag-Wörterbuch gespeichert sind.

24. Die Vorrichtung nach Anspruch 22, wobei ein Wert von „1“ für ein Bit in einem Objektabbildungsvektor die Anwesenheit einer Objekt-ID in dem Gruppensatz angibt, und ein Wert „0“ für ein Bit in einem Objektabbildungsvektor das Fehlen der Objekt-ID in dem Gruppensatz angibt.

25. Die Vorrichtung nach Anspruch 22, wobei die Anzahl individueller Datenwerte in einem beliebigen Gruppensatz gleich der Summe der Werte jedes Bits des Objektabbildungsvektors ist, der diesem Gruppensatz entspricht.

26. Die Vorrichtung nach Anspruch 22, wobei der eine oder die mehreren Gruppensätze mindestens zwei Gruppensätze umfassen, und wobei der eine oder die mehreren Speicher ferner Befehle darin gespeichert haben, die, wenn sie von mindestens einem des einen oder mehreren Prozessoren ausgeführt werden, mindestens einen der einen oder mehreren Prozessoren veranlassen, um:

einen Befehl zur Vereinigung der Daten aus zwei oder mehr der Gruppensätze in einen kombinierten Gruppensatz zu empfangen; und

eine logische ODER-Operation an den Objektabbildungsvektoren auszuführen, die den zwei oder mehr Gruppensätzen entsprechen, um einen kombinierten Objektabbildungsvektor zu erzeugen.

27. Die Vorrichtung nach Anspruch 26, wobei die Anzahl individueller Datenwerte in dem kombinierten Gruppensatz gleich der Summe der Werte jedes der Bits des kombinierten Objektabbildungsvektors ist.

28. Die Vorrichtung nach Anspruch 22, wobei die Tabelle partitioniert und über mehrere Rechenknoten verteilt ist, und wobei die Spalte aus Daten eine erste Spaltenpartition in einem Satz aus Spaltenpartitionen ist, wobei der Bereich eine erste Bereichspartition in einem Satz aus Bereichspartitionen ist, wobei die eine oder die mehreren Objekt-IDs eine oder mehrere erste Partitionsobjekt-IDs und die Objektabbildungsvektoren ein oder mehrere erste Partitionsobjektabbildungsvektoren sind, wobei der eine oder die mehreren Speicher ferner Befehle darin gespeichert haben, die, wenn sie von mindestens einem des einen oder den mehreren Prozessoren ausgeführt werden, mindestens einen des einen oder mehreren Prozessoren veranlassen, um:

die erste Bereichspartition mit einer oder mehreren anderen Bereichspartitionen in dem Satz aus Bereichspartitionen zu kombinieren, um einen globalen Bereich zu erzeugen;

einen oder mehrere der eindeutigen Datenwerte in dem globalen Bereich in Zeichen zu übersetzen, indem jeder des einen oder mehreren eindeutigen Datenwerte auf eine globale Objekt-ID abgebildet wird, um eine oder mehrere globale Objekt-IDs zu erzeugen;

jede der einen oder mehreren ersten Partitionsobjekt-IDs auf eine entsprechende globale Objekt-ID in der einen oder den mehreren globalen Objekt-IDs abzubilden durch Kreuzreferenzierung eines gemeinsamen eindeutigen Datenwertes; und

jeden des einen oder der mehreren ersten Partitionsobjektabbildungsvektoren in einen oder mehrere globale Objektabbildungsvektoren zu übersetzen auf der Grundlage der Abbildung einer oder mehrerer erster Partitionsobjekt-IDs auf eine oder mehrere globale Objekt-IDs.

29. Die Vorrichtung nach Anspruch 28, wobei der eine oder die mehreren Speicher ferner Befehle darin gespeichert haben, die, wenn sie von mindestens einem des einen oder mehreren Prozessoren ausgeführt werden, mindestens einen des einen oder den mehreren Prozessoren veranlassen, um:

eine logische ODER-Operation an dem einen oder den mehreren globalen Objektabbildungsvektoren, die der ersten Partition entsprechen, und an einem oder mehreren zweiten globalen Objektabbildungsvektoren auszuführen, die einer zweiten Partition entsprechen, um einen oder mehrere kombinierte globale Objektabbildungsvektoren zu erzeugen.

30. Die Vorrichtung nach Anspruch 29, wobei die Summe der Werte jedes Bits jedes der kombinierten globalen Objektabbildungsvektoren für die erste Partition und die zweite Partition die Gesamtzahl individueller Elemente in einem speziellen Gruppensatz ist, der diesen globalen Objektabbildungsvektoren entspricht.

31. Die Vorrichtung nach Anspruch 28, wobei die globalen Objektabbildungsvektoren für jede Partition in einem globalen Datenwürfel gespeichert sind, wobei der globale Datenwürfel eine Objektabbildungsvektorenabmessung, eine Gruppensatzabmessung und eine Partitionsabmessung aufweist.

32. Die Vorrichtung nach Anspruch 22, wobei die Anzahl an Gruppensätzen zumindest teilweise durch die Anzahl individueller Datenwerte in einer zweiten Spalte aus Daten in der Tabelle bestimmt ist.

33. Die Vorrichtung nach Anspruch 22, wobei die Gruppensätze in Bezug auf eine oder mehrere zweite Spalten aus Daten in der Tabelle definiert sind.

34. Die Vorrichtung nach Anspruch 22, jede Objekt-ID auf jeden eindeutigen Datenwert in chronologischer Reihenfolge des Auftretens der eindeutigen Datenwerte abgebildet ist, so dass die erste Objekt-ID auf den ersten eindeutigen angetroffenen Datenwert abgebildet ist.

35. Die Vorrichtung nach Anspruch 34, wobei die eine oder die mehreren Objekt-IDs eine oder mehrere globale Objekt-IDs sind und der eine oder die mehreren Objektabbildungsvektoren einer oder mehrere vorher existierende globale Objektabbildungsvektoren sind, und wobei der eine oder die mehreren Speicher ferner

Befehle darin gespeichert haben, die, wenn sie von mindestens einem des einen oder den mehreren Prozessoren ausgeführt werden, mindestens einen des einen oder den mehreren Prozessoren veranlassen, um:
 eine Aktualisierung mit einem oder mehreren neuen Objektabbildungsvektoren zu empfangen, die einer oder mehreren neuen Objekt-IDs entsprechen, wobei jede neue Objekt-ID einem eindeutigen Datenwert entspricht;
 eine neue globale Objekt-ID jedem eindeutigen Datenwert zuzuweisen, der nicht bereits in dem Bereich repräsentiert ist; und
 einen oder mehrere globale Objektabbildungsvektoren zu erzeugen, die dem einen oder den mehreren neuen Objektabbildungsvektoren entsprechen, wobei die Länge jedes globalen Objektabbildungsvektors gleich der Gesamtzahl globaler Objekt-IDs ist, und wobei der Wert jedes Bits in jedem globalen Objektabbildungsvektor die Anwesenheit oder das Fehlen einer anderen globalen Objekt-ID in einem entsprechenden Gruppensatz angibt.

36. Die Vorrichtung nach Anspruch 25, wobei der eine oder die mehreren Speicher ferner Befehle darin gespeichert haben, die, wenn sie von mindestens einem des einen oder mehreren Prozessoren ausgeführt werden, mindestens einen des einen oder mehreren Prozessoren veranlassen, um:
 die vorher existierenden globalen Objektabbildungsvektoren durch Anhängen einer Anzahl an Nullen an das Ende jedes vorher existierenden globalen Objektabbildungsvektors zu aktualisieren, wobei die Anzahl an Nullen gleich der Anzahl eindeutiger Datenwerte in der Aktualisierung ist, die nicht bereits in dem Bereich repräsentiert waren.

37. Die Vorrichtung nach Anspruch 22, wobei jede Objekt-ID auf jeden eindeutigen Datenwert in lexikalischer Reihenfolge der eindeutigen Datenwerte abgebildet ist, so dass die erste Objekt-ID auf den eindeutigen Datenwert abgebildet ist, der als erstes in der lexikalischen Reihenfolge der eindeutigen Datenwerte auftritt.

38. Die Vorrichtung nach Anspruch 22, wobei die numerische Platzierung jedes Bits in einem Objektabbildungsvektor der Objekt-ID entspricht, deren Anwesenheit oder Fehlen in dem Gruppensatz das Bit repräsentiert.

39. Die Vorrichtung nach Anspruch 22, wobei die in Zeichen übersetzte Spalte aus Daten nach dem Empfang einer Abfrage gespeichert ist.

40. Die Vorrichtung nach Anspruch 22, wobei die in Zeichen übersetzten Spalte aus Daten vor Empfang einer Abfrage gespeichert ist.

41. Die Vorrichtung nach Anspruch 22, wobei die Abfrage einen Summenbefehl enthält, der jeden des einen oder mehreren Gruppensätze betrifft, und wobei der eine oder die mehreren Speicher ferner Befehle darin gespeichert haben, die, wenn sie von mindestens einem des einen oder den mehreren Prozessoren ausgeführt werden, mindestens einen des einen oder mehreren Prozessoren veranlassen, um:
 für jeden der einen oder mehreren Gruppensätze die eindeutigen Datenwerte zu summieren, die den Objekt-IDs entsprechen, die als anwesend in dem Objektabbildungsvektor, der diesem Gruppensatz entspricht, angegeben sind.

42. Die Vorrichtung nach Anspruch 22, wobei der Abfrage einen Mittelwertbefehl enthält, der jeden des einen oder der mehreren Gruppensätze betrifft, und wobei der eine oder die mehreren Speicher ferner Befehle darin gespeichert haben, die, wenn sie von mindestens einem des einen oder mehreren Prozessoren ausgeführt werden, mindestens einen des einen oder mehreren Prozessoren veranlassen, um:
 für jeden des einen oder der mehreren Gruppensätze den Mittelwert der eindeutigen Datenwerte zu berechnen, die den Objekt-IDs entsprechen, die als anwesend in dem Objektabbildungsvektor, der diesem Gruppensatz entspricht, angegeben sind.

43. Mindestens ein nicht-flüchtiges computerlesbares Medium, das computerlesbare Befehle speichert, die, wenn sie von einer oder mehreren Recheneinrichtungen ausgeführt werden, mindestens eine der einen oder mehreren Recheneinrichtungen veranlassen, um:
 eine in Zeichen übersetzte Spalte aus Daten in einer Tabelle zu speichern, wobei die in Zeichen übersetzte Spalte aus Daten erzeugt wird, durch Abbilden jedes eindeutigen Datenwertes in einem Bereich, der einer Spalte aus Daten entspricht, auf eine Objekt-ID, und durch Ersetzen jedes Datenwertes in der Spalte durch eine entsprechende Objekt-ID, um die Spalte aus Zeichen übersetzter Daten, die eine oder mehrere Objekt-IDs enthält, zu erzeugen;
 eine Abfrage zu empfangen, die die Spalte aus Daten betrifft, wobei die Abfrage einen oder mehrere Gruppensätze zur Gruppierung der Daten definiert, die in Reaktion auf die Abfrage abgerufen werden; und

einen Objektabbildungsvektor für jeden Gruppensatz in dem einen oder den mehreren Gruppensätzen zu erzeugen, wobei die Länge jedes Objektabbildungsvektors gleich der Gesamtanzahl an Objekt-IDs für den Bereich ist, und wobei der Wert jedes Bits in jedem Objektabbildungsvektor die Anwesenheit oder das Fehlen einer anderen Objekt-ID in dem entsprechenden Gruppensatz angibt.

44. Das mindestens eine nicht-flüchtige computerlesbare Medium nach Anspruch 43, wobei die Abbildungen zwischen jeder der Objekt-IDs und den eindeutigen Datenwerten in einen Nachschlag-Wörterbuch gespeichert sind

45. Das mindestens eine nicht-flüchtige computerlesbare Medium nach Anspruch 43, wobei ein Wert von „1“ für ein Bit in einem Objektabbildungsvektor die Anwesenheit einer Objekt-ID in dem Gruppensatz angibt, und ein Wert „0“ für ein Bit in einem Objektabbildungsvektor das Fehlen der Objekt-ID in dem Gruppensatz angibt.

46. Das mindestens eine nicht-flüchtige computerlesbare Medium nach Anspruch 43, wobei die Anzahl individueller Datenwerte in einem beliebigen Gruppensatz gleich der Summe der Werte jedes Bits des Objektabbildungsvektors ist, der diesem Gruppensatz entspricht.

47. Das mindestens eine nicht-flüchtige computerlesbare Medium nach Anspruch 43, wobei der eine oder die mehreren Gruppensätze mindestens zwei Gruppensätze umfassen, und wobei das mindestens eine nicht-flüchtige computerlesbare Medium ferner Befehle aufweist, die, wenn sie von mindestens einer der einen oder mehreren Recheneinrichtungen ausgeführt werden, mindestens eine der einen oder mehreren Recheneinrichtungen veranlassen, um:

einen Befehl zur Vereinigung der Daten aus zwei oder mehr der Gruppensätze in einen kombinierten Gruppensatz zu empfangen; und

eine logische ODER-Operation an den Objektabbildungsvektoren auszuführen, die den zwei oder mehr Gruppensätzen entsprechen, um einen kombinierten Objektabbildungsvektor zu erzeugen.

48. Das mindestens eine nicht-flüchtige computerlesbare Medium nach Anspruch 47, wobei die Anzahl individueller Datenwerte in dem kombinierten Gruppensatz gleich der Summe der Werte jedes der Bits des kombinierten Objektabbildungsvektors ist.

49. Das mindestens eine nicht-flüchtige computerlesbare Medium nach Anspruch 43, wobei die Tabelle partitioniert und über mehrere Rechenknoten verteilt ist, und wobei die Spalte aus Daten eine erste Spaltenpartition in einem Satz aus Spaltenpartitionen ist, wobei der Bereich eine erste Bereichspartition in einem Satz aus Bereichspartitionen ist, wobei die eine oder die mehreren Objekt-IDs eine oder mehrere erste Partitionsobjekt-IDs und die Objektabbildungsvektoren ein oder mehrere erste Partitionsobjektabbildungsvektoren sind, wobei das mindestens eine nicht-flüchtige computerlesbare Medium ferner Befehle aufweist, die, wenn sie von mindestens einer der einen oder den mehreren Recheneinrichtungen ausgeführt werden, mindestens eine der einen oder mehreren Recheneinrichtungen veranlassen, um:

die erste Bereichspartition mit einer oder mehreren anderen Bereichspartitionen in dem Satz aus Bereichspartitionen zu kombinieren, um einen globalen Bereich zu erzeugen;

einen oder mehrere der eindeutigen Datenwerte in dem globalen Bereich in Zeichen zu übersetzen, indem jeder des einen oder mehreren eindeutigen Datenwerte auf eine globale Objekt-ID abgebildet wird, um eine oder mehrere globale Objekt-IDs zu erzeugen;

jede der einen oder mehreren ersten Partitionsobjekt-IDs auf eine entsprechende globale Objekt-ID in der einen oder den mehreren globalen Objekt-IDs abzubilden durch Kreuzreferenzierung eines gemeinsamen eindeutigen Datenwertes; und

jeden des einen oder der mehreren ersten Partitionsobjektabbildungsvektoren in einen oder mehrere globale Objektabbildungsvektoren zu übersetzen auf der Grundlage der Abbildung einer oder mehrerer erster Partitionsobjekt-IDs auf eine oder mehrere globale Objekt-IDs.

50. Das mindestens eine nicht-flüchtige computerlesbare Medium nach Anspruch 49, wobei das mindestens eine nicht-flüchtige computerlesbare Medium ferner Befehle aufweist, die, wenn sie von mindestens einer der einen oder mehreren Recheneinrichtungen ausgeführt werden, mindestens eine der einen oder mehreren Recheneinrichtungen veranlassen, um:

eine logische ODER-Operation an dem einen oder den mehreren globalen Objektabbildungsvektoren, die der ersten Partition entsprechen, und an einem oder mehreren zweiten globalen Objektabbildungsvektoren auszuführen, die einer zweiten Partition entsprechen, um einen oder mehrere kombinierte globale Objektabbildungsvektoren zu erzeugen.

51. Das mindestens eine nicht-flüchtige computerlesbare Medium nach Anspruch 50, wobei die Summe der Werte jedes Bits jedes der kombinierten globalen Objektabbildungsvektoren für die erste Partition und die zweite Partition die Gesamtzahl individueller Elemente in einem speziellen Gruppensatz ist, der diesen globalen Objektabbildungsvektoren entspricht.

52. Das mindestens eine nicht-flüchtige computerlesbare Medium nach Anspruch 49, wobei die globalen Objektabbildungsvektoren für jede Partition in einem globalen Datenwürfel gespeichert sind, wobei der globale Datenwürfel eine Objektabbildungsvektorenabmessung, eine Gruppensatzabmessung und eine Partitionsabmessung aufweist.

53. Das mindestens eine nicht-flüchtige computerlesbare Medium nach Anspruch 43, wobei die Anzahl an Gruppensätzen zumindest teilweise durch die Anzahl individueller Datenwerte in einer zweiten Spalte aus Daten in der Tabelle bestimmt ist.

54. Das mindestens eine nicht-flüchtige computerlesbare Medium nach Anspruch 43, wobei die Gruppensätze in Bezug auf eine oder mehrere zweite Spalten aus Daten in der Tabelle definiert sind.

55. Das mindestens eine nicht-flüchtige computerlesbare Medium nach Anspruch 43, wobei jede Objekt-ID auf jeden eindeutigen Datenwert in chronologischer Reihenfolge des Auftretens der eindeutigen Datenwerte abgebildet ist, so dass die erste Objekt-ID auf den ersten eindeutigen angetroffenen Datenwert abgebildet ist.

56. Das mindestens eine nicht-flüchtige computerlesbare Medium nach Anspruch 55, wobei die eine oder die mehreren Objekt-IDs eine oder mehrere globale Objekt-IDs sind und der eine oder die mehreren Objektabbildungsvektoren einer oder mehrere vorher existierende globale Objektabbildungsvektoren sind, und wobei das mindestens eine nicht-flüchtige computerlesbare Medium ferner Befehle aufweist, die, wenn sie von mindestens einer der einen oder mehreren Recheneinrichtungen ausgeführt werden, mindestens eine der einen oder mehreren Recheneinrichtungen veranlassen, um:

eine Aktualisierung mit einem oder mehreren neuen Objektabbildungsvektoren zu empfangen, die einer oder mehreren neuen Objekt-IDs entsprechen, wobei jede neue Objekt-ID einem eindeutigen Datenwert entspricht; eine neue globale Objekt-ID jedem eindeutigen Datenwert zuzuweisen, der nicht bereits in dem Bereich repräsentiert ist; und

einen oder mehrere globale Objektabbildungsvektoren zu erzeugen, die dem einen oder den mehreren neuen Objektabbildungsvektoren entsprechen, wobei die Länge jedes globalen Objektabbildungsvektors gleich der Gesamtzahl globaler Objekt-IDs ist, und wobei der Wert jedes Bits in jedem globalen Objektabbildungsvektor die Anwesenheit oder das Fehlen einer anderen globalen Objekt-ID in einem entsprechenden Gruppensatz angibt.

57. Das mindestens eine nicht-flüchtige computerlesbare Medium nach Anspruch 56, wobei das mindestens eine nicht-flüchtige computerlesbare Medium ferner Befehle aufweist, die, wenn sie von mindestens einer der einen oder mehreren Recheneinrichtungen ausgeführt werden, mindestens eine der einen oder mehreren Recheneinrichtungen veranlassen, um:

die vorher existierenden globalen Objektabbildungsvektoren durch Anhängen einer Anzahl an Nullen an das Ende jedes vorher existierenden globalen Objektabbildungsvektors zu aktualisieren, wobei die Anzahl an Nullen gleich der Anzahl eindeutiger Datenwerte in der Aktualisierung ist, die nicht bereits in dem Bereich repräsentiert waren.

58. Das mindestens eine nicht-flüchtige computerlesbare Medium nach Anspruch 43, wobei jede Objekt-ID auf jeden eindeutigen Datenwert in lexikalischer Reihenfolge der eindeutigen Datenwerte abgebildet ist, so dass die erste Objekt-ID auf den eindeutigen Datenwert abgebildet ist, der als erstes in der lexikalischen Reihenfolge der eindeutigen Datenwerte auftritt.

59. Das mindestens eine nicht-flüchtige computerlesbare Medium nach Anspruch 43, wobei die numerische Platzierung jedes Bits in einem Objektabbildungsvektor der Objekt-ID entspricht, deren Anwesenheit oder Fehlen in dem Gruppensatz das Bit repräsentiert.

60. Das mindestens eine nicht-flüchtige computerlesbare Medium nach Anspruch 43, wobei die in Zeichen übersetzte Spalte aus Daten nach dem Empfang einer Abfrage gespeichert wird.

61. Das mindestens eine nicht-flüchtige computerlesbare Medium nach Anspruch 43, wobei die in Zeichen übersetzten Spalte aus Daten vor Empfang einer Abfrage gespeichert wird.

62. Das mindestens eine nicht-flüchtige computerlesbare Medium nach Anspruch 43, wobei die Abfrage einen Summenbefehl enthält, der jeden des einen oder mehreren Gruppensätze betrifft, und wobei das mindestens eine nicht-flüchtige computerlesbare Medium ferner Befehle aufweist, die, wenn sie von mindestens einer der einen oder mehreren Recheneinrichtungen ausgeführt werden, mindestens eine der einen oder mehreren Recheneinrichtungen veranlassen, um:

für jeden der einen oder mehreren Gruppensätze die eindeutigen Datenwerte zu summieren, die den Objekt-IDs entsprechen, die als anwesend in dem Objektabbildungsvektor, der diesem Gruppensatz entspricht, angegeben sind.

63. Das mindestens eine nicht-flüchtige computerlesbare Medium nach Anspruch 43, wobei der Abfrage einen Mittelwertbefehl enthält, der jeden des einen oder der mehreren Gruppensätze betrifft, und wobei das mindestens eine nicht-flüchtige computerlesbare Medium ferner Befehle aufweist, die, wenn sie von mindestens einer der einen oder mehreren Recheneinrichtungen ausgeführt werden, mindestens eine der einen oder mehreren Recheneinrichtungen veranlassen, um:

für jeden des einen oder der mehreren Gruppensätze den Mittelwert der eindeutigen Datenwerte zu berechnen, die den Objekt-IDs entsprechen, die als anwesend in dem Objektabbildungsvektor, der diesem Gruppensatz entspricht, angegeben sind.

Es folgen 18 Seiten Zeichnungen

Anhängende Zeichnungen

Kundenname	Warenhaus-ID	Ausgegebener Betrag
Bill	1	10
Bill	2	5
George	1	11
George	2	11
Max	2	21
Larry	1	31
Larry	2	18

Fig. 1A

Warenhaus-ID	Anzahl an Besuchen
1	3
2	4

Fig. 1B

Warenhaus-ID	Ausgegebener Betrag
1	52
2	55

Fig. 1C

Warenhaus-ID	Mittlerer ausgegebener Betrag	Anzahl an Besuchen
1	17 1/3	3
2	13 3/4	4

Fig. 1D

Warenhaus-ID	Ausgegebenes Minimum
1	10
2	5

Fig. 1E

Warenhaus-ID	Ausgegebenes Maximum
1	31
2	21

Fig. 1F

Warenhaus-ID	Anzahl individueller Kunden
1	3
2	4

Fig. 1G

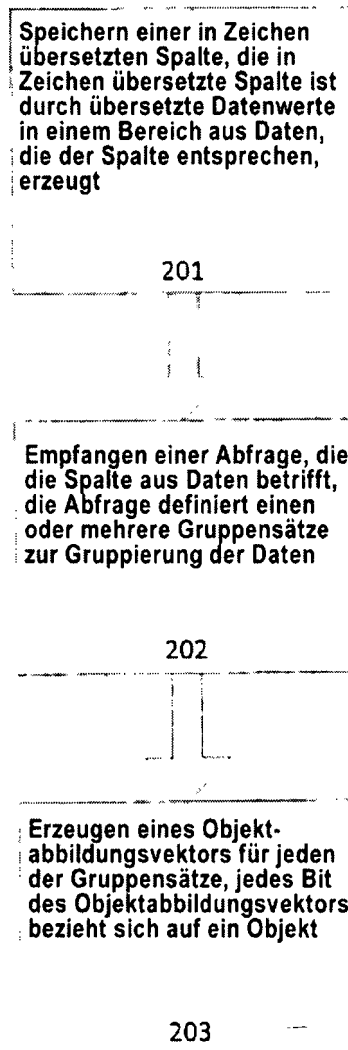


Fig. 2A

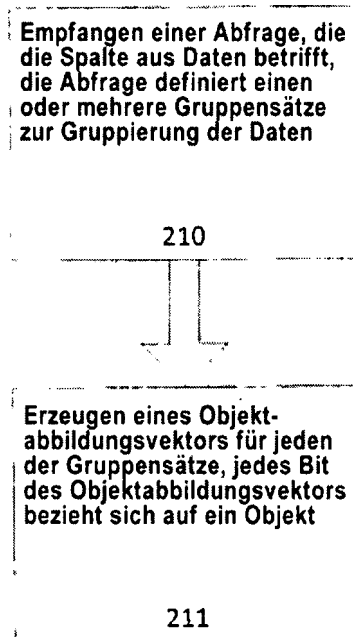


Fig. 2B

Tupel-ID	Woche	Warenhaus-ID	KundenID	ProduktID	Menge	Ausgegeben
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	40
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	40
19	201003	1001	165	11	10	40
20	201003	1003	167	20	30	30

Fig. 3A

D_Week		D_WhouseID		D_CustomerID		D_ProductID		D_Amount		D_AmountGiven	
EID	Wert	EID	Wert	EID	Wert	EID	Wert	EID	Wert	EID	Wert
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

Fig. 3B

D_Woche		D_WarenhausID		D_KundenID		D_ProduktID		D_Menge		D_Ausgegeben	
EID	Wert	EID	Wert	EID	Wert	EID	Wert	EID	Wert	EID	Wert
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

TupelID	Woche	WarenhausID	KundenID	ProduktID	Menge	Ausgegeben
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	2
7	1	2	3	2	1	3
8	1	2	3	4	3	4
9	1	2	4	2	2	2
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

Fig. 3C

TupelID	Woche	Warenhaus-ID	CD_KundenID
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

Fig. 4A

400

TupelID	Woche	CD_CD_KundenID
1	1	1
2	2	2
3	3	1

Fig. 4B

410

TupelID	Woche	Sum_CD_KundenID
1	1	4
2	2	5
3	3	2

Fig. 4C

420

TupelID	Woche	Warenhaus-ID	KundenID
1	1	1	051100
2	1	2	050011
3	2	1	051101
4	2	2	050010
5	2	3	050100
6	3	1	051000
7	3	3	050100

Fig. 5A

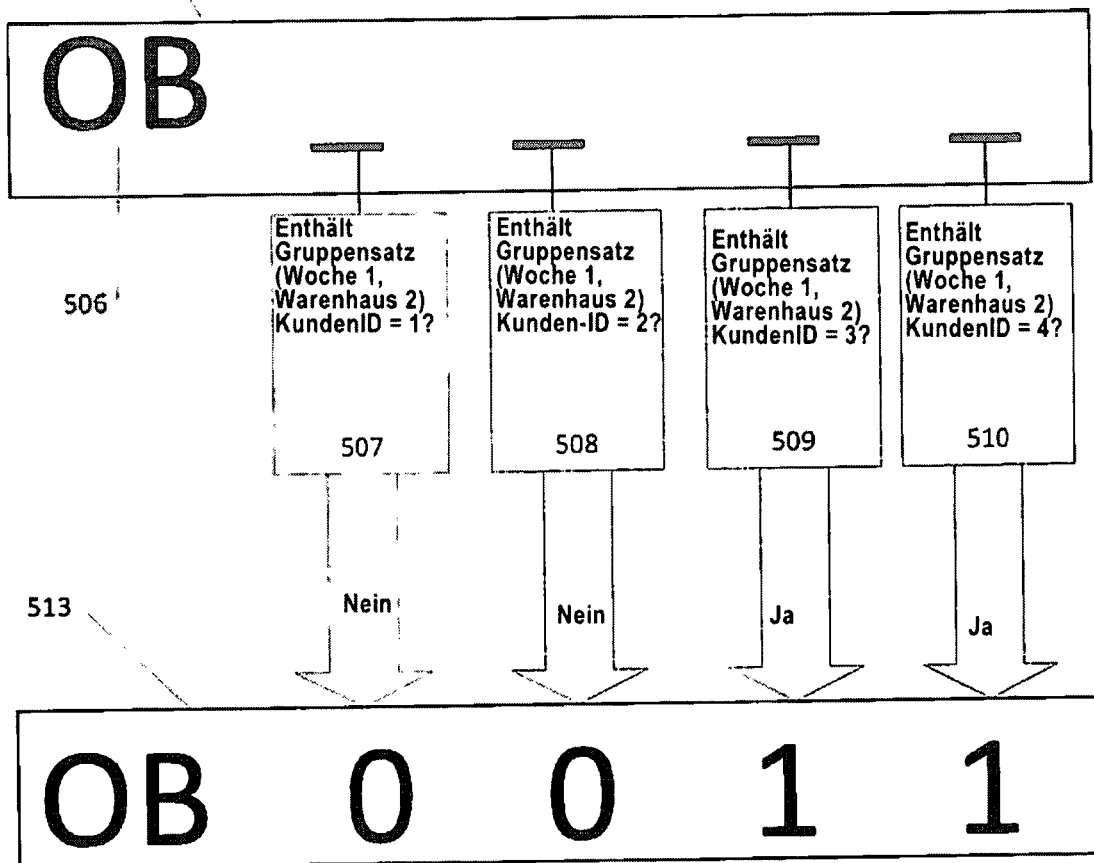


Fig. 5B

520

Lexikalisch geordnete Liste aus eindeutigen in Datenwerten D _ KundenID
165
167
268
301

Fig. 5C

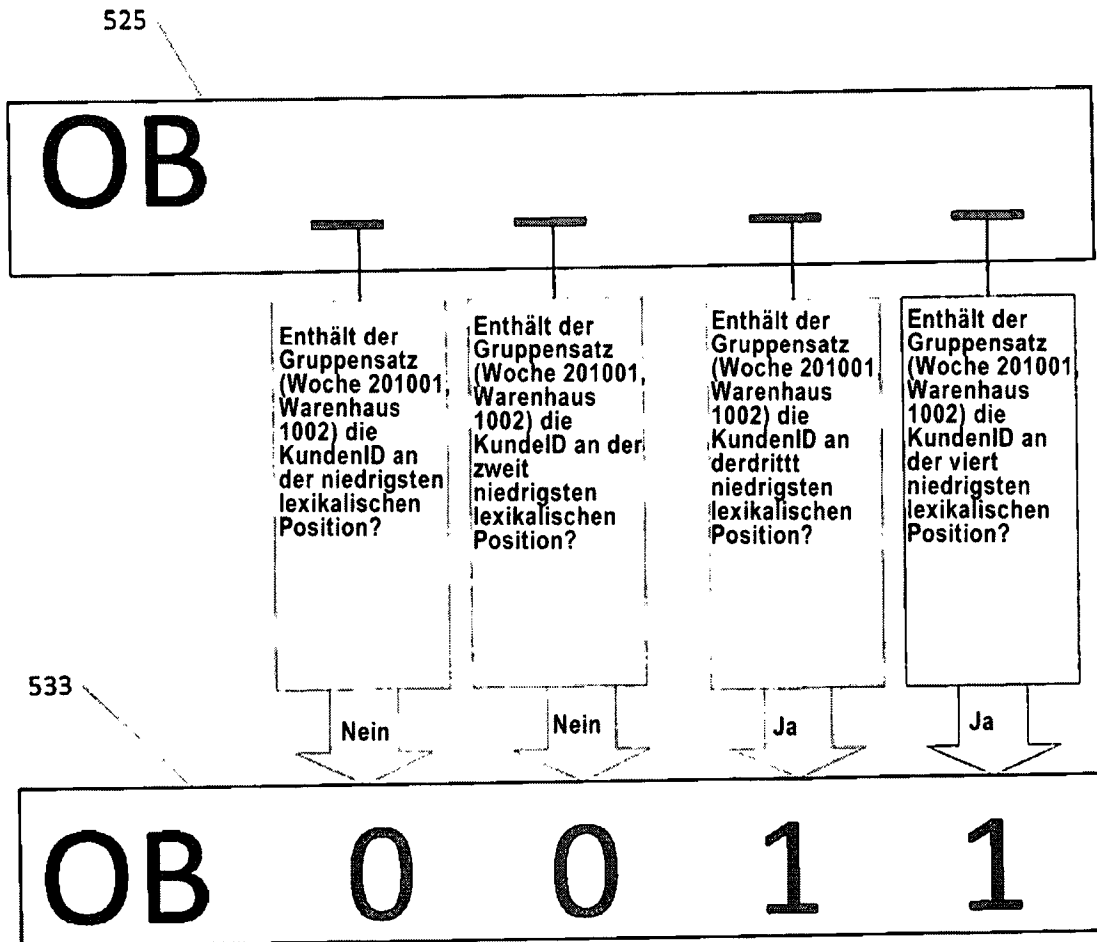


Fig. 5D

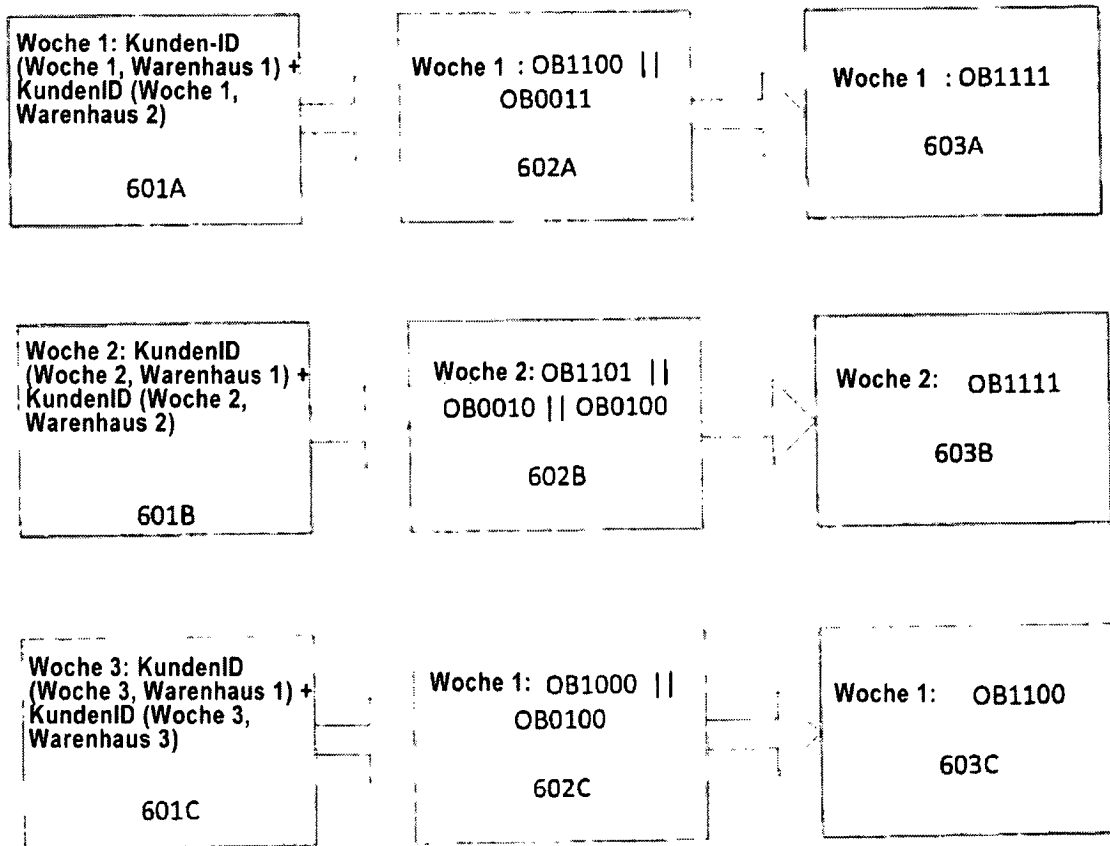


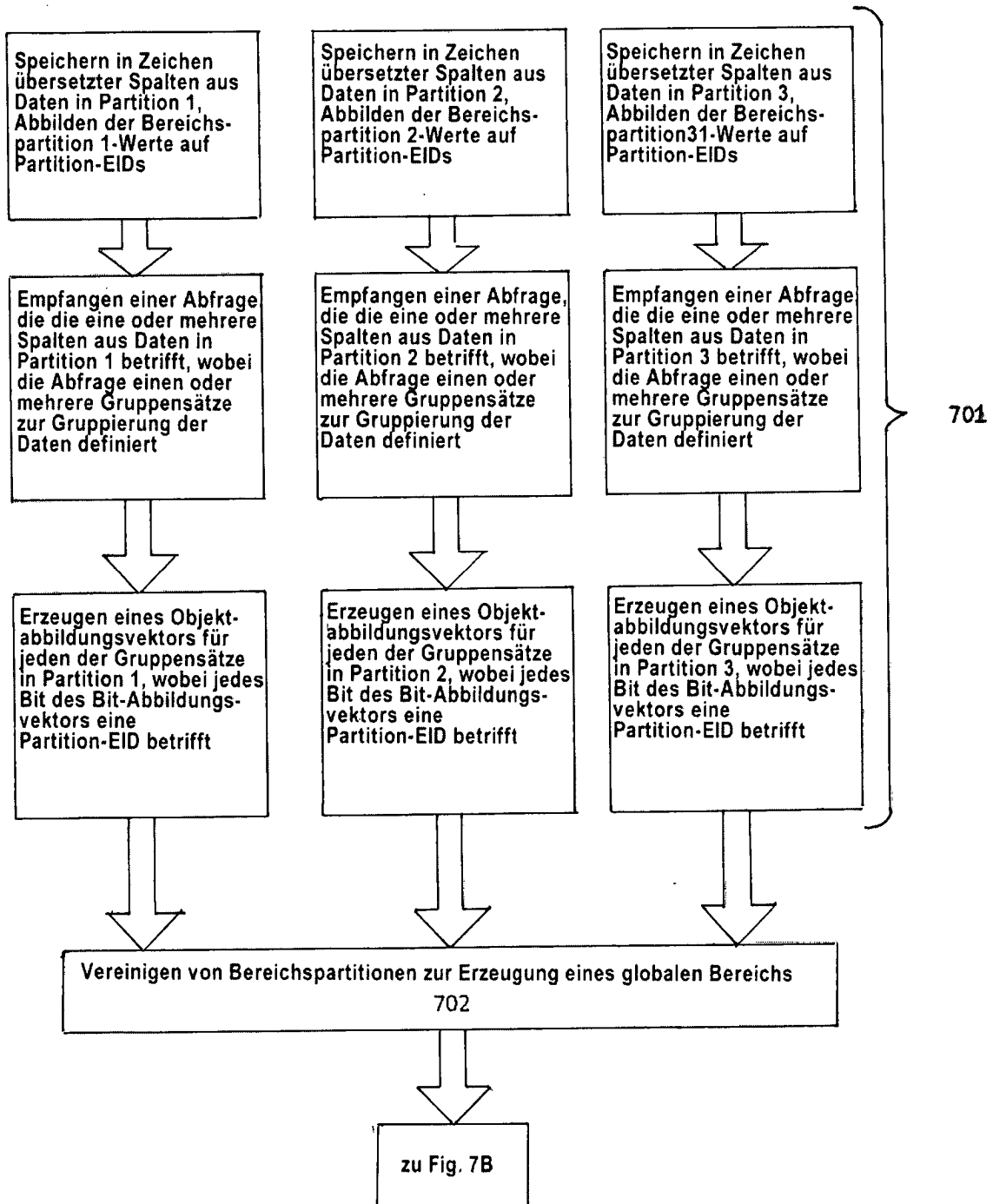
Fig. 6A

TupelID..	Woche	EMV_CD_KundenID
1	1	4
2	2	4
3	3	2

Fig. 6B

610

Fig. 7A



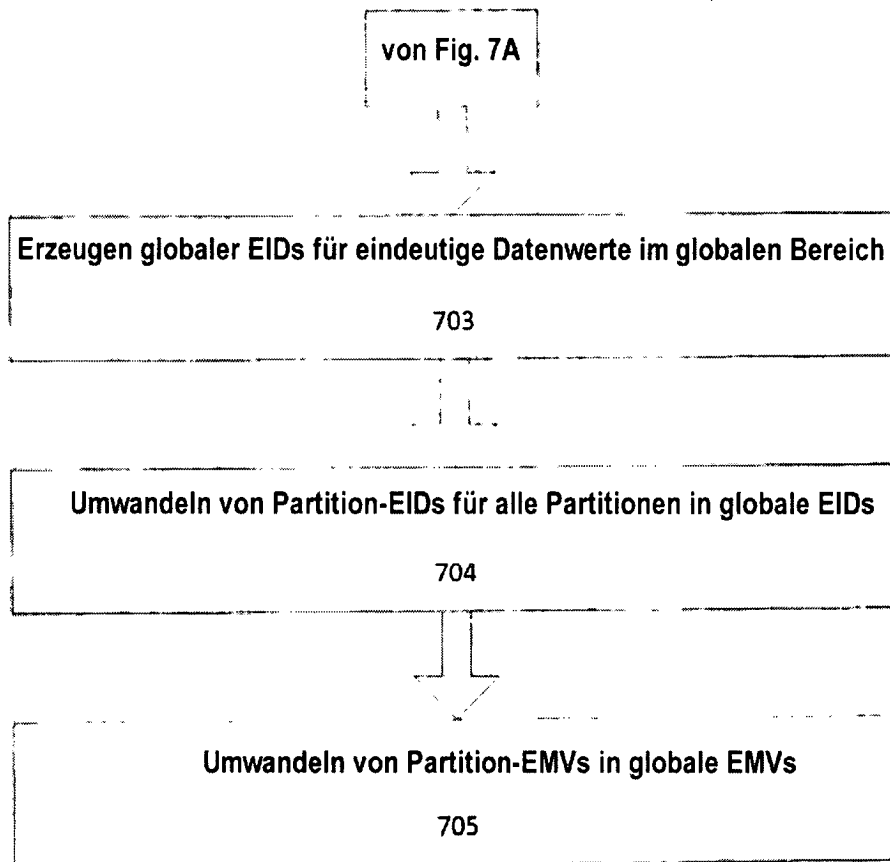


Fig. 7B

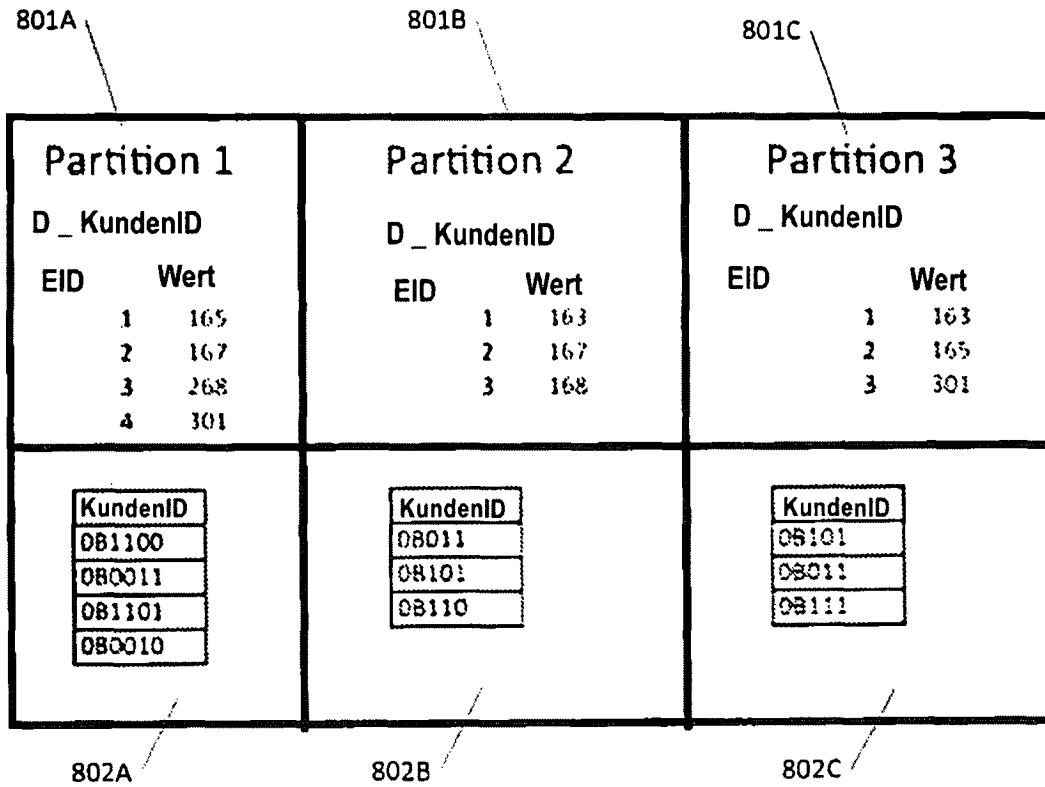


Fig. 8A

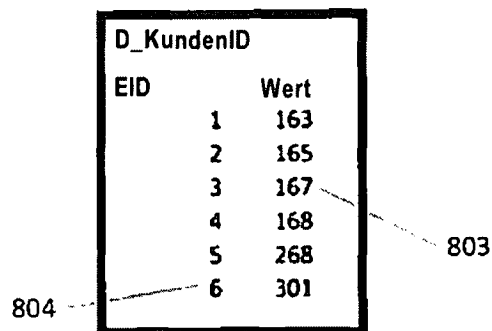


Fig. 8B

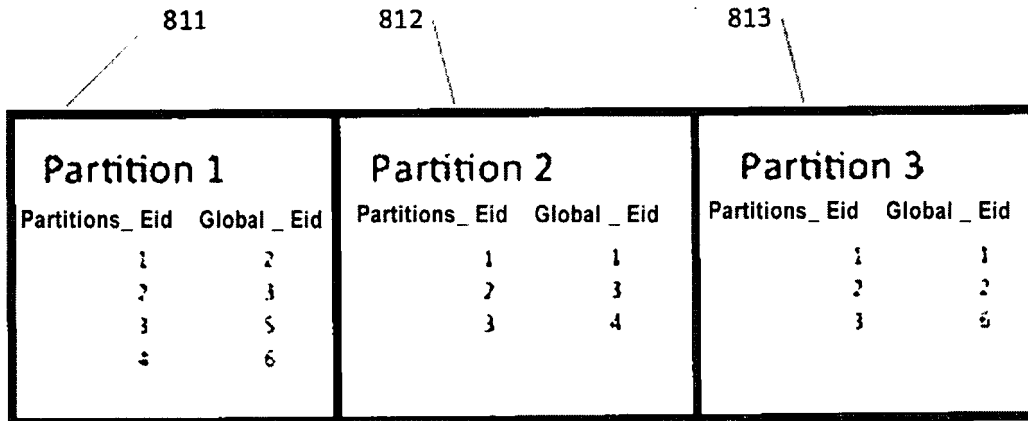


Fig. 8C

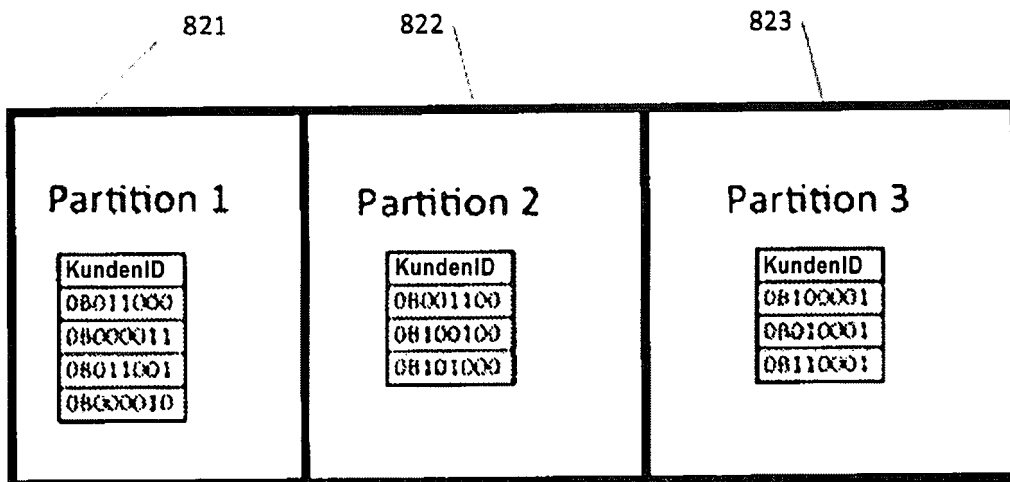


Fig. 8D

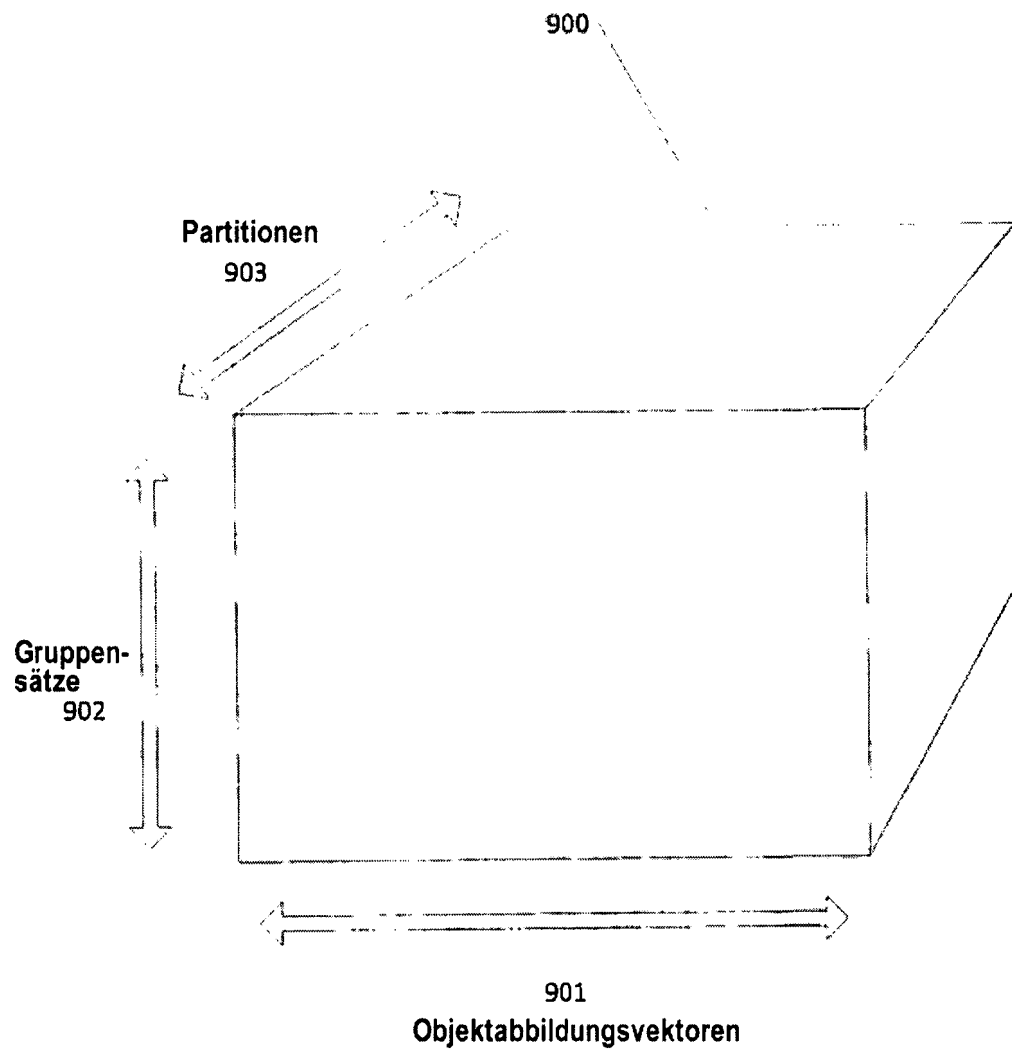


Fig. 9

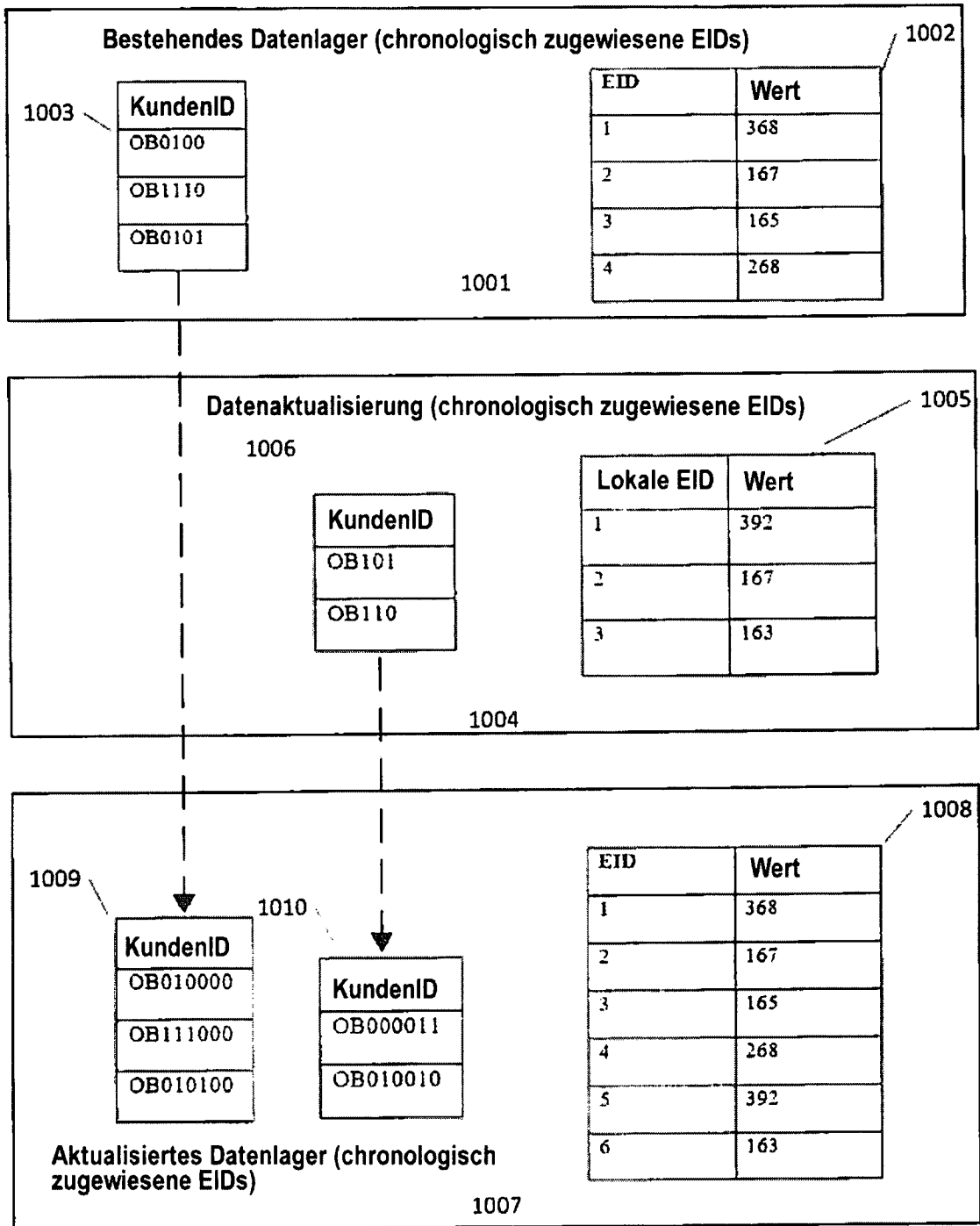


Fig. 10

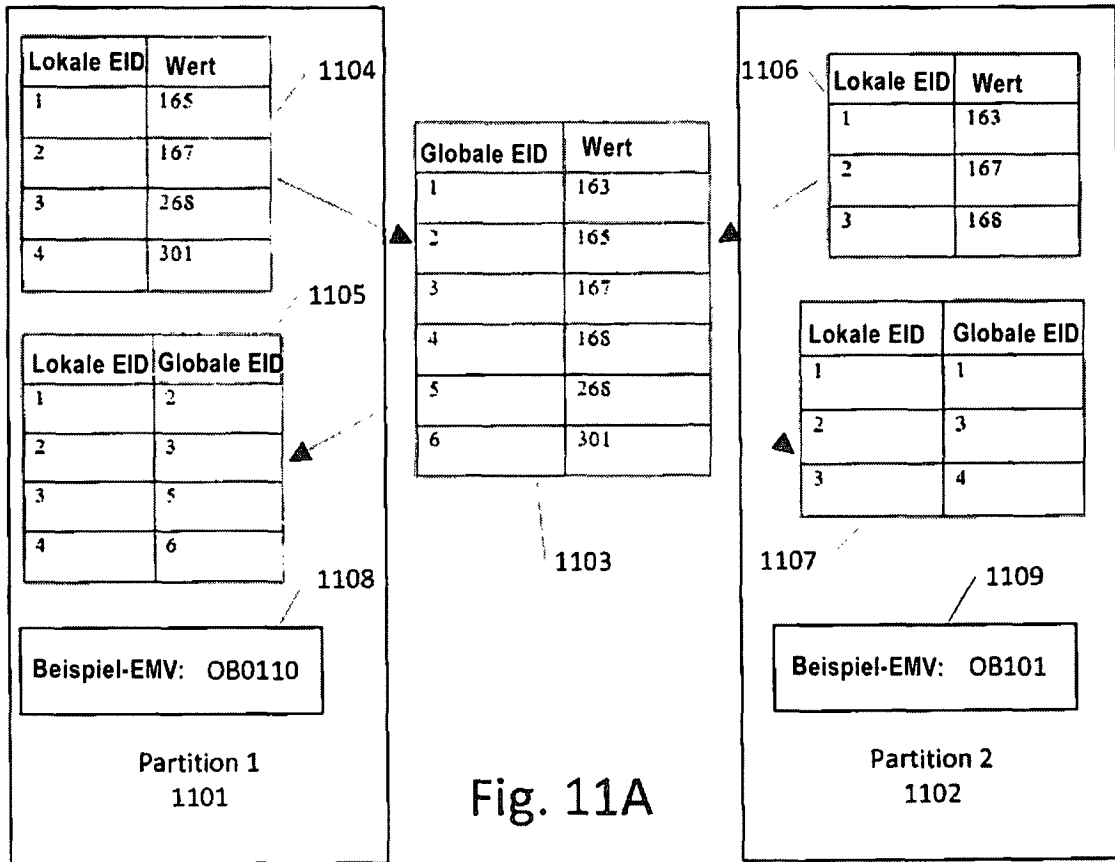


Fig. 11A

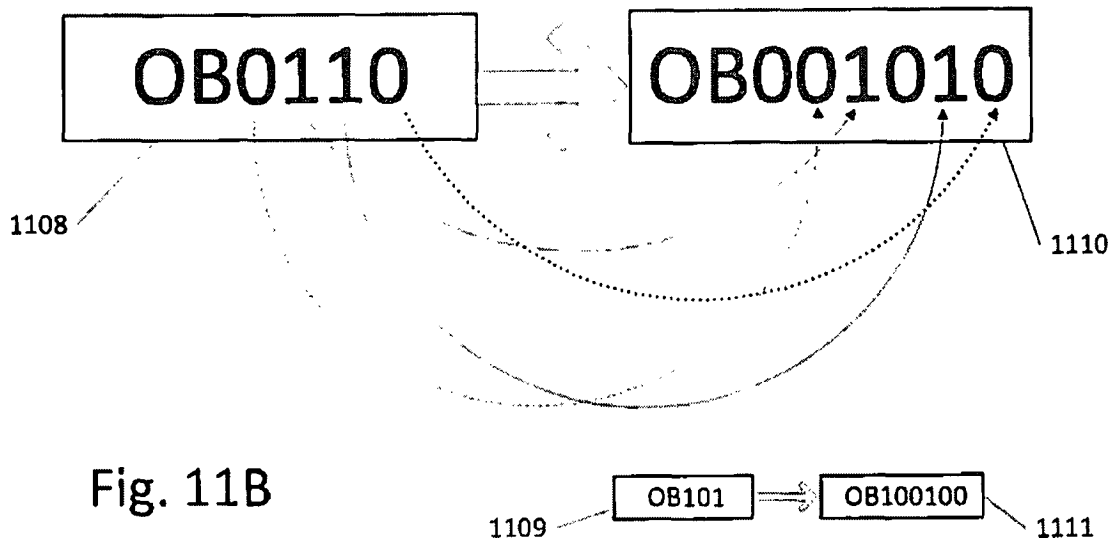


Fig. 11B

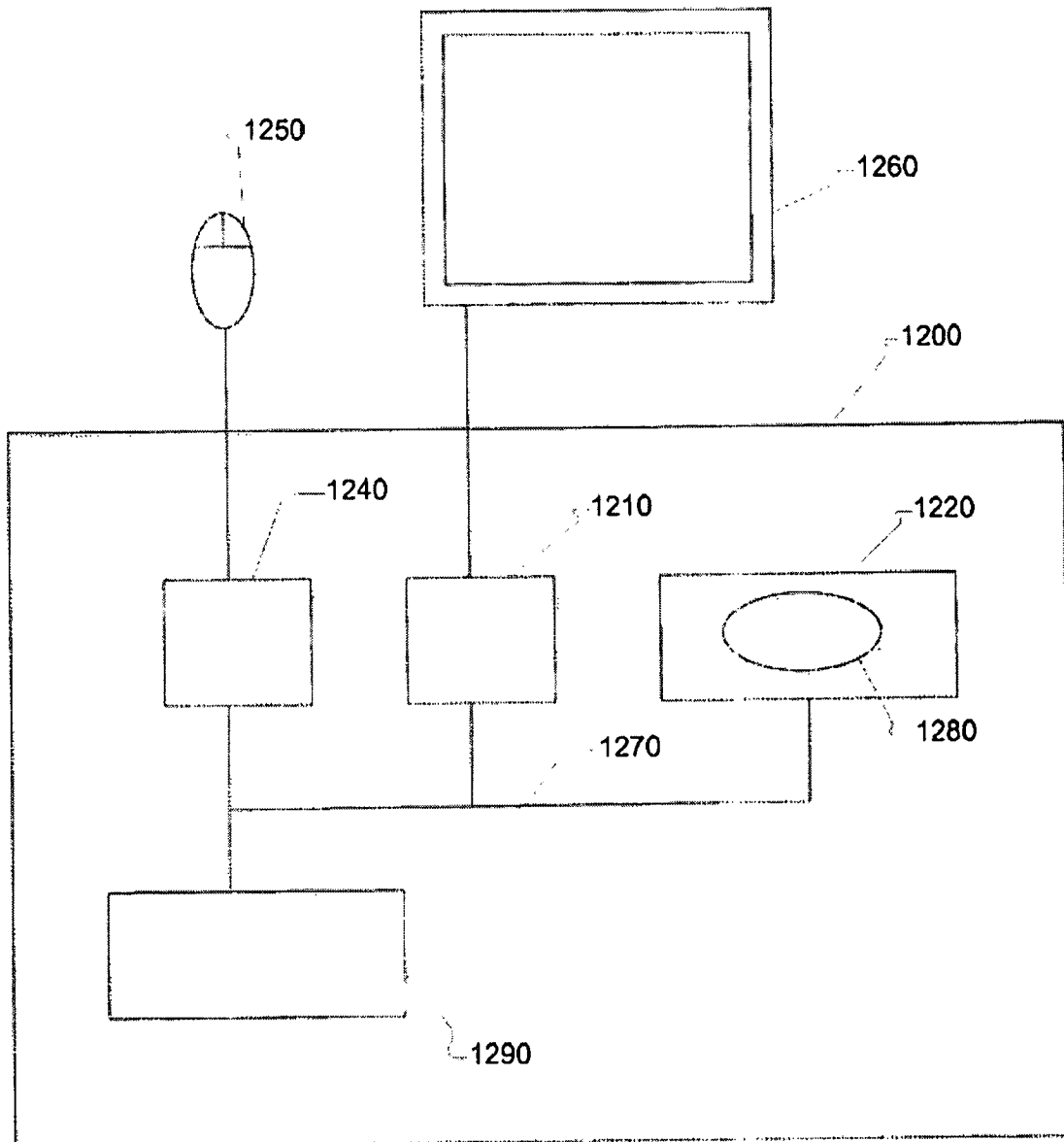


Fig. 12