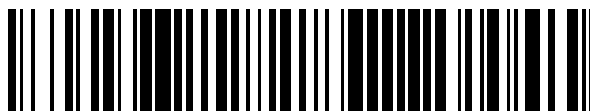


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 713 097**

51 Int. Cl.:

G06F 17/30 (2006.01)

G06F 12/08 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **03.07.2009 E 09164490 (6)**

97 Fecha y número de publicación de la concesión europea: **23.01.2019 EP 2146292**

54 Título: **Método y aparato para extraer información de una base de datos**

30 Prioridad:

18.07.2008 SE 0801708

18.07.2008 US 81761

45 Fecha de publicación y mención en BOPI de la traducción de la patente:

17.05.2019

73 Titular/es:

QLIKTECH INTERNATIONAL AB (100.0%)

Scheelevägen 24-26

223 63 Lund, SE

72 Inventor/es:

WOLGÉ, HÅKAN

74 Agente/Representante:

TOMAS GIL, Tesifonte Enrique

ES 2 713 097 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Método y aparato para extraer información de una base de datos

5 Campo técnico

[0001] La presente invención se refiere a técnicas para extraer información de una base de datos y, en particular, a técnicas que implican una cadena secuencial de cálculos principales que comprende un primer cálculo principal que opera un primer elemento de selección en un conjunto de datos que representan la base de datos para producir un primer resultado, y un segundo cálculo principal que opera un segundo elemento de selección en el primer resultado para producir un segundo resultado.

Estado de la técnica

15 [0002] Con frecuencia, se desea extraer información específica de una base de datos y, especialmente, resumir una gran cantidad de datos en la base de datos y presentar los datos resumidos a un usuario de una forma clara. Dicho procesamiento de datos normalmente se lleva a cabo por un ordenador y puede requerir una capacidad de memoria y una potencia de tratamiento significativas por parte del ordenador. El procesamiento de datos puede tener como objetivo la creación de una estructura de datos grande comúnmente conocida como un cubo multidimensional, a la que el usuario puede a su vez acceder para explorar los datos de la base de datos, por ejemplo visualizando datos seleccionados en las tablas dinámicas o gráficamente en gráficos 2D y 3D. Un ejemplo de un algoritmo eficaz para crear dicho cubo multidimensional se conoce de la patente US705862.

25 [0003] Este algoritmo del estado de la técnica, como muchos otros algoritmos que operan en datos en una base de datos, implica una cadena secuencial de cálculos principales, en los que el resultado de un cálculo principal se usa como datos de entrada por un cálculo principal posterior. Por ejemplo, en el contexto de la patente US7058621, el registro de datos de la base de datos se lee en la memoria principal, a partir de lo cual un usuario puede seleccionar una o más variables y, opcionalmente, un valor o rango de valores para cada variable, provocando de esta forma que el algoritmo extraiga un subconjunto correspondiente del registro de datos en la base de datos. El subconjunto extraído forma un resultado intermedio. El cubo multidimensional se calcula a continuación evaluando una función matemática seleccionada en el subconjunto extraído, donde la evaluación de la función matemática se lleva a cabo basándose en un conjunto seleccionado de variables de cálculo, y donde las dimensiones del cubo se obtienen de un conjunto seleccionado de variables de clasificación.

35 [0004] Aunque el algoritmo del estado de la técnica es eficaz, todavía puede necesitar llevar a cabo un gran número de operaciones para crear el cubo multidimensional, especialmente si se deben analizar grandes cantidades de datos. En dichas situaciones, el algoritmo puede establecer requisitos indeseablemente altos en el hardware de procesamiento y/o presentar un tiempo de cálculo más largo de lo preferible.

40 [0005] La patente US2006/0230024 se refiere a un método para una infraestructura caché basada en el contexto para habilitar la consulta en el subconjunto acerca de un objeto en caché. En respuesta a la detección de una consulta a un contexto raíz de un árbol de contexto, se atraviesa el árbol hasta un contexto original de un subcontexto que corresponde con el par de nombre y valor, que es identificado por un usuario en la consulta. Si el contexto original almacena en caché todos resultados de la consulta, los resultados de la consulta se iteran y los pares de nombre y valor restantes se ignoran. Sin embargo, si el contexto original no almacena en caché todos los resultados de la consulta, la etapa de atravesar se repite para el siguiente contexto original del subcontexto hasta que se encuentra un contexto raíz. Si se encuentra un contexto raíz, se expide una consulta a la base de datos para el par de nombre y valor y el resultado de la consulta a la base de datos se almacena en caché en un contexto nuevo.

50 Resumen

[0006] Un objetivo de la invención es superar al menos parcialmente una o más de las limitaciones del estado de la técnica identificadas anteriormente.

55 [0007] Este y otros objetivos, que aparecerán en la descripción siguiente, se consiguen al menos parcialmente mediante un método, un medio legible por ordenador y un aparato según las reivindicaciones independientes, donde los ejemplos de realización de los mismos se definen por las reivindicaciones dependientes.

60 [0008] Un primer aspecto de la invención es un método implementado por ordenador para extraer información de una base de datos, donde dicho método comprende una cadena secuencial de cálculos principales que comprende un primer cálculo principal que opera un primer elemento de selección en un conjunto de datos que representa la base de datos para producir un resultado intermedio, y un segundo cálculo principal que opera un segundo elemento de selección en el resultado intermedio para producir un resultado final, donde dicho método comprende además la recuperación del resultado final mediante las etapas de:

65

(a) calcular un primer valor de identificador de selección (ID1) como una huella digital estadísticamente única generada por una función hash de al menos el primer elemento de selección (S1);

(b) buscar, en los objetos de la estructura de datos, el primer valor de identificador de selección (ID1) y, si se encuentra el primer valor de identificador de selección (ID1), localizar y recuperar un primer identificador de resultado (ID2) almacenado con el primer valor de identificador de selección (ID1) como objetos asociados en una iteración precedente;

(c) si el primer identificador de resultado (ID2) se encuentra en la subetapa (b),

calcular un segundo valor de identificador de selección (ID3) como una huella digital estadísticamente única generada por una función hash de al menos el segundo elemento de selección (S2) y el primer identificador de resultado (ID2) recuperado, y

buscar, en los objetos de la estructura de datos, el segundo valor de identificador de selección (ID3) y, si se encuentra el segundo valor de identificador de selección (ID3), localizar y recuperar un resultado final (R2) almacenado con el segundo valor de identificador de selección (ID3) como objetos asociados en una iteración precedente;

(d) si no se encuentra el primer identificador de resultado (ID2) en la subetapa (b),

ejecutar el primer cálculo principal (P1) para producir el resultado intermedio (R1) y el primer valor de identificador de resultado (ID2) como una huella digital generada por una función hash del resultado intermedio (R1),

almacenar el primer valor de identificador de selección (ID1) y el primer valor de identificador de resultado (ID2) como objetos asociados en la estructura de datos; y

almacenar el primer valor de identificador de resultado (ID2) y el resultado intermedio (R1) como objetos asociados en la estructura de datos,

calcular un segundo valor de identificador de selección (ID3) como una huella digital estadísticamente única generada por una función hash del primer valor de identificador de resultado (ID2) y el segundo elemento de selección (S2), y

buscar en los objetos de la estructura de datos basándose en el segundo valor de identificador de selección (ID3) y, si se encuentra el segundo valor de identificador de selección (ID3), localizar y recuperar un resultado final (R2) almacenado con el segundo valor de identificador de selección (ID3) como objetos asociados en una iteración precedente;

(e) si no se encuentra el resultado final (R2) en la subetapa (c) o (d),

buscar en los objetos de la estructura de datos basándose en el primer valor de identificador de resultado (ID2);

(f) si no se encuentra el primer valor de identificador de resultado (ID2) en la subetapa (e),

ejecutar el primer cálculo principal (P1) para producir el resultado intermedio (R1) y el primer valor de identificador de resultado (ID2) como una huella digital generada por una función hash del resultado intermedio (R1),

almacenar el primer valor de identificador de resultado (ID2) y el resultado intermedio (R1) como objetos asociados en la estructura de datos, y

ejecutar el segundo cálculo principal (P2) para producir el resultado final (R2) y almacenar el segundo valor de identificador de selección (ID3) y el resultado final (R2) como objetos asociados en la estructura de datos; y

(g) si se encuentra el primer valor de identificador de resultado (ID2) en la subetapa (e),

recuperar el resultado intermedio (R1) almacenado con el primer valor de identificador de resultado (ID2) como objetos asociados en una iteración precedente, y

ejecutar el segundo cálculo principal (P2) para producir el resultado final (R2) y almacenar el segundo valor de identificador de selección (ID3) y el resultado final (R2) como objetos asociados en la estructura de datos.

[0009] De esta forma, en el método según el primer aspecto, el primer y el segundo resultado se guardan en caché en la memoria informática y quedan disponibles para su reutilización en iteraciones posteriores del método, reduciendo así la necesidad de ejecutar el primer y/o segundo cálculo principal para la extracción de la información. La reutilización puede implicar calcular el primer y/o segundo valor de identificador de selección durante una iteración posterior y acceder a la estructura de datos para recuperar potencialmente el primer y/o segundo resultado.

[0010] Un segundo aspecto de la invención es un medio legible por ordenador con un programa informático almacenado que, cuando lo ejecuta un ordenador, es apto para llevar a cabo el método según el primer aspecto.

[0011] Un tercer aspecto de la invención es un aparato para extraer información de una base de datos, donde dicho aparato incluye medios para ejecutar una cadena secuencial de cálculos principales que comprende un

primer cálculo principal que opera un primer elemento de selección en un conjunto de datos que representa la base de datos para producir un primer resultado, y un segundo cálculo principal que opera un segundo elemento de selección en el primer resultado para producir un segundo resultado, donde dicho aparato comprende además medios para recuperar el resultado final mediante los pasos de:

5

(a) calcular un primer valor de identificador de selección (ID1) como una huella digital estadísticamente única generada por una función hash de al menos el primer elemento de selección (S1);

10

(b) buscar, en los objetos de la estructura de datos, el primer valor de identificador de selección (ID1) y, si se encuentra el primer valor de identificador de selección (ID1), localizar y recuperar un primer identificador de resultado (ID2) almacenado con el primer valor de identificador de selección (ID1) como objetos asociados en una iteración precedente;

(c) si se encuentra el primer identificador de resultado (ID2) en la subetapa (b),

15

calcular un segundo valor de identificador de selección (ID3) como una huella digital estadísticamente única generada por una función hash de al menos el segundo elemento de selección (S2) y el primer identificador de resultado (ID2) recuperado, y

20

buscar, en los objetos de la estructura de datos, el segundo valor de identificador de selección (ID3) y, si se encuentra el segundo valor de identificador de selección (ID3), localizar y recuperar un resultado final (R2) almacenado con el segundo valor de identificador de selección (ID3) como objetos asociados en una iteración precedente;

(d) si no se encuentra el primer identificador de resultado (ID2) en la subetapa (b),

25

ejecutar el primer cálculo principal (P1) para producir el resultado intermedio (R1) y el primer valor de identificador de resultado (ID2) como una huella digital generada por una función hash del resultado intermedio (R1),

almacenar el primer valor de identificador de selección (ID1) y el primer valor de identificador de resultado (ID2) como objetos asociados en la estructura de datos; y

30

almacenar el primer valor de identificador de resultado (ID2) y el resultado intermedio (R1) como objetos asociados en la estructura de datos,

calcular un segundo valor de identificador de selección (ID3) como una huella digital estadísticamente única generada por una función hash del primer valor de identificador de resultado (ID2) y el segundo elemento de selección (S2), y

35

buscar en los objetos de la estructura de datos basándose en el segundo valor de identificador de selección (ID3) y, si se encuentra el segundo valor de identificador de selección (ID3), localizar y recuperar un resultado final (R2) almacenado con el segundo valor de identificador de selección (ID3) como objetos asociados en una iteración precedente;

(e) si el resultado final (R2) no se encuentra en la subetapa (c) o (d),

40

buscar en los objetos de la estructura de datos basándose en el primer valor de identificador de resultado (ID2);

(f) si el primer valor de identificador de resultado (ID2) no se encuentra en la subetapa (e),

45

ejecutar el primer cálculo principal (P1) para producir el resultado intermedio (R1) y el primer valor de identificador de resultado (ID2) como una huella digital generada por una función hash del resultado intermedio (R1),

almacenar el primer valor de identificador de resultado (ID2) y el resultado intermedio (R1) como objetos asociados en la estructura de datos, y

50

ejecutar el segundo cálculo principal (P2) para producir el resultado final (R2) y almacenar el segundo valor de identificador de selección (ID3) y el resultado final (R2) como objetos asociados en la estructura de datos; y

(g) si el primer valor de identificador de resultado (ID2) se encuentra en la subetapa (e),

55

recuperar el resultado intermedio (R1) almacenado con el primer valor de identificador de resultado (ID2) como objetos asociados en una iteración precedente, y

ejecutar el segundo cálculo principal (P2) para producir el resultado final (R2) y almacenar el segundo valor de identificador de selección (ID3) y el resultado final (R2) como objetos asociados en la estructura de datos.

60

[0012] El aparato del tercer aspecto comparte las ventajas del método del primer aspecto y puede comprender características adicionales que corresponden a cualquiera de los ejemplos de realización anteriormente descritos en relación con el primer aspecto.

65

[0013] Otros objetivos, características, aspectos y ventajas de la presente invención aparecerán en la siguiente descripción detallada, las reivindicaciones adjuntas y los dibujos.

Breve descripción de los dibujos

[0014] Los ejemplos de realización de la invención se describirán a continuación con más detalle con referencia a los dibujos esquemáticos de acompañamiento, donde los mismos números de referencia se utilizan para identificar los elementos correspondientes.

La Fig. 1 ilustra un proceso que implica una cadena de cálculos para extraer información de una base de datos, donde los identificadores y los resultados se almacenan selectivamente y se recuperan de una memoria informática.

La Fig. 2 ilustra un ejemplo de un proceso como se muestra en la Fig. 1.

La Fig. 3 ilustra otro ejemplo de un proceso como se muestra en la Fig. 1.

La Fig. 4 ilustra una forma de realización del proceso de la Fig. 1.

La Fig. 5 ilustra otro ejemplo de un proceso como se muestra en la Fig. 1.

La Fig. 6 es un diagrama de flujo que ejemplifica el proceso de la Fig. 5.

La Fig. 7 es una visión de conjunto del proceso de la Fig. 5 implementada en un contexto específico

La Fig. 8 es un diagrama de bloques de un entorno de ordenador para implementar los ejemplos de realización de la invención.

Descripción detallada de las formas de realización ejemplares

[0015] La presente invención se refiere a técnicas para extraer información a partir de una base de datos. Para una mayor facilidad de comprensión, algunos principios subyacentes se discutirán en primer lugar en relación con un ejemplo general. Más adelante, se comentarán diferentes aspectos, características y ventajas en relación con una implementación específica.

GENERAL

[0016] La Fig. 1 ilustra un ejemplo de un proceso implementado en el ordenador para extraer información de una base de datos DB, que puede almacenarse o no almacenarse de forma externa al ordenador que implementa el proceso. El proceso de extracción incluye la extracción de un conjunto o campo de datos inicial RO de la base de datos DB, por ejemplo leyendo el conjunto de datos inicial RO en la memoria principal (por ejemplo RAM) del ordenador. El conjunto de datos inicial RO puede incluir todo el contenido de la base de datos DB o un subconjunto de la misma.

[0017] El proceso de la Figura 1 incluye una secuencia de procedimientos de cálculo principales P1, P2 que operan para generar un resultado final R2 basado en el conjunto de datos inicial RO. En concreto, un primer procedimiento P1 opera en el conjunto de datos inicial RO para producir un resultado intermedio R1, y el segundo procedimiento P2 opera en el resultado intermedio para producir el resultado final R2.

[0018] Un primer elemento de selección S1, que puede originarse o no a partir de una entrada del usuario, controla el primer procedimiento P1. De forma similar, un segundo elemento de selección S2, que puede originarse o no a partir de una entrada del usuario, controla el segundo procedimiento P2. Cada elemento de selección S1, S2 puede incluir cualquier combinación de variables y para funciones matemáticas que definen los datos de entrada al respectivo procedimiento, es decir, el conjunto de datos RO y el resultado intermedio R1, respectivamente.

[0019] La Fig. 1 también indica que el proceso de extracción interactúa con una memoria informática 10 (habitualmente memoria RAM o caché), con el primer y el segundo procedimiento P1, P2 operando para almacenar elementos de datos en la memoria 10 y recuperar los elementos de datos de la memoria 10. En el ejemplo ilustrado, el primer procedimiento P1 opera para memorizar y recuperar los identificadores, generalmente designados por ID, y los resultados intermedios R1; y el segundo procedimiento P2 opera para memorizar y recuperar los identificadores, generalmente designados por ID, los resultados intermedios R1 y los resultados finales R2. En lo sucesivo, el procedimiento de almacenamiento de identificadores y resultados en la memoria informática 10 también se denominará "almacenamiento en caché".

[0020] Los diferentes identificadores se generan habitualmente mediante los procedimientos P1, P2 en función de uno o más parámetros del proceso, tales como otro identificador y/o un elemento de selección S1, S2 y/o un resultado R1, R2. Las diferentes funciones pueden usarse o no para generar diferentes identificadores. La/las función/funciones para generar un identificador puede(n) ser un algoritmo de hash que genere una huella digital del/de los parámetro(s) de proceso pertinente(s). La función/funciones se configura(n) adecuadamente de manera que cada combinación de valores de parámetro únicos tiene como resultado un valor identificador que es único entre todos los valores identificadores que se generan para todos los diferentes identificadores en el proceso. En este contexto, "único" no solo incluye valores de identificador teóricamente únicos, sino también valores de identificador estadísticamente únicos. Un ejemplo no limitativo de dicha función es un algoritmo de hash que genera una huella digital de al menos 256 bits.

[0021] En un proceso ilustrado más adelante en la Fig. 2, el primer procedimiento P1 se configura para que calcule un primer valor de identificador de selección ID 1 en función del primer elemento de selección S1, es decir $ID1=f(S1)$ y el segundo procedimiento P2 se configura para que calcule un segundo valor de identificador de selección ID3 en función del segundo elemento de selección S2 y el resultado intermedio R1, es decir $ID3=f(S2, R1)$. El primer procedimiento P1 también se configura para que almacene ID1 y el resultado intermedio R1 como objetos asociados en una estructura de datos 12 en la memoria informática, y el segundo procedimiento P2 se configura para que almacene ID3 y R2 como objetos asociados en la estructura de datos 12. De este modo, la estructura de datos 12 en la memoria informática 10 se configura para que almacene un conjunto heterogéneo de objetos, es decir, objetos de diferentes tipos.

[0022] Este proceso permite una reducción del tiempo de respuesta en el proceso de extracción y/o una reducción de los requisitos de tratamiento del ordenador que implementa el proceso de extracción reduciendo la necesidad de ejecutar los procedimientos de cálculo principales P1, P2 para calcular el resultado intermedio R1 y el resultado final R2, respectivamente. Por ejemplo, el proceso de extracción se puede configurar para que use la estructura de datos 12, siempre que sea posible, para encontrar el resultado final R2 basado en el primer elemento de selección S1 y el segundo elemento de selección S2. Así, cuando el proceso descubre una necesidad de calcular el resultado final R2, basado en S1 y S2, puede generar $ID1=f(S1)$ y acceder a la estructura de datos 12 basada en ID1. Si se ha usado un primer elemento de selección S1 idéntico con el primer procedimiento P1 precedente, es posible encontrar el valor generado de ID1 en la estructura de datos 12 y que este se asocie con el resultado intermedio correspondiente R1. De este modo, el resultado intermedio R1 se puede recuperar a partir de la estructura de datos 12 en lugar de calculándolo mediante el procedimiento P1. Si el resultado intermedio R1 no se encuentra en la estructura de datos 12, el proceso puede hacer que el primer procedimiento P1 calcule el resultado intermedio R1. Además, después de la obtención del resultado intermedio R1, el proceso puede generar $ID3=f(R1; S2)$ y acceder a la estructura de datos 12 basada en ID3. Nuevamente, si la misma operación ha sido ejecutada por el procedimiento anterior, es posible que el valor generado de ID3 se encuentre en la estructura de datos 12 y se asocie al resultado final correspondiente R2. Por lo tanto, el resultado final R2 se puede recuperar de la estructura de datos 12 en lugar de calculándolo mediante el procedimiento P2.

[0023] En el proceso ilustrado más adelante en la Fig. 3, el primer procedimiento P1 se configura posteriormente para calcular un primer valor de identificador de resultado ID2 en función del resultado intermedio R1. El primer procedimiento P1 también se configura para que almacene ID 1 e ID2 como objetos asociados en la estructura de datos 12 y para que almacene ID2 y el resultado intermedio R1 como objetos asociados en la estructura de datos 12.

[0024] Esta forma de realización del proceso permite una reducción del tamaño de la memoria informática requerida por el proceso, ya que cada resultado intermedio R1 se almacena solo una vez en la estructura de datos 12, aunque dos o más primeros elementos de selección S1 produzcan resultados intermedios idénticos R1. Esta forma de realización es particularmente pertinente cuando los resultados intermedios R1 son altos, que es con frecuencia el caso cuando se procesa información de bases de datos.

[0025] El cálculo del primer valor de identificador de resultado ID2 también permite una forma de realización, ilustrada en la Fig. 4, en la que el resultado intermedio R1 se representa mediante el primer valor de identificador de resultado ID2 en el cálculo del segundo valor de identificador de selección ID3, es decir $ID3=f(ID2, S2)$.

[0026] Esta forma de realización reduce la necesidad de almacenar el resultado intermedio R1 en la estructura de datos 12, ya que se puede recuperar el resultado final R2 de la estructura de datos 12 basándose en ID3, que se genera basándose en ID2, no el resultado intermedio R1. Esto permite calcular eficazmente el resultado final R2, aunque el resultado intermedio R1 se ha eliminado de la estructura de datos 12. Por ejemplo, el proceso se puede configurar para que use la estructura de datos 12, siempre que sea posible, para encontrar el resultado final R2 basándose en el primer elemento de selección S1 y el segundo elemento de selección S2. Así, cuando el proceso descubre una necesidad de calcular el resultado final R2, basándose en S1 y S2, puede generar $ID1=f(S1)$ y acceder a la estructura de datos 12 basándose en ID1 para recuperar ID2 asociado con ello, si previamente se ha usado un primer elemento de selección idéntico S1 con el primer procedimiento P1. A continuación, el proceso puede generar $ID3=f(ID2; S2)$ y acceder a la estructura de datos 12 basándose en ID3 para recuperar el resultado final R2 asociado con ello, si el segundo procedimiento P2 ha operado en un resultado intermedio idéntico R1 y un segundo elemento de selección idéntico S2 previamente. En este ejemplo, el resultado final R2 puede recuperarse así de la estructura de datos 12 aunque se haya eliminado el resultado intermedio R1.

[0027] En un proceso ilustrado en la Fig. 5, el primer procedimiento P1 se configura posteriormente para que calcule un segundo valor de identificador de resultado ID4 en función del resultado final R2. El segundo procedimiento P2 también se configura para que almacene ID3 e ID4 como objetos asociados en la estructura de datos 12 y para que almacene ID4 y el resultado final R2 como objetos asociados en la estructura de datos 12.

[0028] Este proceso permite reducir el tamaño de la memoria informática requerida por el proceso, ya que cada resultado final R2 se almacena solo una vez en la estructura de datos 12, aunque dos o más segundos elementos de selección S2 produzcan resultados finales idénticos R2. Este proceso es particularmente pertinente cuando los resultados finales R2 son altos.

5

[0029] Hasta ahora, se ha supuesto que la base de datos DB, y por tanto el conjunto de datos R0, es estática. Si la base de datos es dinámica, puede ser adecuado generar el primer identificador de selección ID1 en función del primer elemento de selección S1 y el conjunto de datos R0, es decir, $ID1=f(S1, R0)$. Con dicha modificación, todos los procesos y formas de realización descritos en relación con la Fig. 1-5 se pueden aplicar igualmente a una base de datos dinámica, es decir, una base de datos que puede cambiar en cualquier momento.

10

[0030] La Fig. 6 es un diagrama de flujo que ilustra una implementación de ejemplificación del proceso mostrado en la Fig. 5, adaptado para que opere en una base de datos dinámica. El proceso comienza con la entrada del conjunto de datos R0 (etapa 600), el primer elemento de selección S1 (etapa 602) y el segundo elemento de selección S2 (604). A continuación, se genera un valor del primer identificador de selección ID1 en función de S1 y R0 (etapa 606). Se lleva a cabo una búsqueda en la estructura de datos basada en ID1 (etapa 608). Si se encuentra el valor de ID1 en la estructura de datos, es decir, se ha almacenado en caché en una iteración precedente, el proceso recupera el valor del primer identificador de resultado ID2 asociado con ello (etapa 610) y procede a la etapa 612.

15

20

[0031] Si no se encuentra el valor de ID1 en la estructura de datos en la etapa 608, el proceso provoca que el primer procedimiento P1 calcule R1, operando S1 en R0 (etapa 614). A continuación, el valor de ID2 se genera en función de R1 (etapa 616), y los valores de ID1, ID2 y R1 se almacenan en la estructura de datos en pares asociados ID1:ID2 e ID2:R1 (etapa 618). Seguidamente, el proceso continúa a la etapa 612.

25

[0032] En la etapa 612, el valor del segundo identificador de selección ID3 se genera en función de S2 e ID2. Después, se lleva a cabo una búsqueda en la estructura de datos basada en ID3 (etapa 620). Si se encuentra el valor de ID3 en la estructura de datos, es decir, se ha almacenado en caché en una iteración precedente, el proceso recupera el valor del segundo identificador de resultado ID4 asociado con ello (etapa 622). Se lleva a cabo una búsqueda adicional en la estructura de datos basada en ID4 (etapa 624). Si se encuentra el valor de ID4 en la estructura de datos, es decir, se ha almacenado en caché en una iteración precedente, el proceso recupera el resultado final R2 asociado con ello (etapa 626).

30

[0033] Si no se encuentra el valor de ID3 en la estructura de datos en la etapa 620, se lleva a cabo una búsqueda adicional en la estructura de datos basada en el valor de ID2 determinado en la etapa 610 o en la etapa 616 (etapa 628). Si se encuentra el valor de ID2 en la estructura de datos, es decir, se ha almacenado en caché en una iteración precedente, el proceso recupera el primer resultado R1 asociado con ello (etapa 630). El proceso provoca a continuación que el segundo procedimiento P2 calcule R2, operando S2 en R1 (etapa 632). Para actualizar la estructura de datos, el proceso también genera el valor de ID4 en función de R2 (etapa 634) y almacena los valores de ID3, ID4 y R2 en la estructura de datos en pares asociados ID3:ID4 e ID4:R2 (etapa 636).

35

40

[0034] Si no se encuentra el valor de ID2 en la estructura de datos en la etapa 628, el proceso provoca que el primer procedimiento P1 calcule R1, operando S1 en R0 (etapa 638), y almacena los valores de ID2 y R1 en la estructura de datos en un par asociado ID2:R1 (etapa 640). El proceso procede a continuación a la etapa 632. Sin embargo, debe tenerse en cuenta que si el resultado intermedio R1 ya se ha calculado en la etapa 614, no es necesario realizar las etapas 628, 630, 638 y 640. En tal caso, si ID3 no se encuentra en la etapa 620, el proceso puede proceder directamente a la etapa 632, donde se provoca que el segundo procedimiento P2 calcule R2, operando S2 en R1.

45

50

[0035] Si no se encuentra el valor de ID4 en la estructura de datos en la etapa 622, el proceso provoca que el segundo procedimiento P2 calcule R2, operando S2 en R1 (etapa 642). Para actualizar la estructura de datos, el proceso también genera el valor de ID4 en función de R2 (etapa 644) y almacena los valores de ID4 y R2 en la estructura de datos en un par asociado ID4:R2 (etapa 646).

55

[0036] El experto entenderá fácilmente que los procesos y ejemplos de realización de las Figs. 2-4 provocan los procesos de almacenamiento y recuperación correspondientes, utilizando sin embargo combinaciones de identificadores diferentes. Para mantener la brevedad de la presentación, estos procesos no se ilustran en las tablas de flujo, sino que se presentan meramente como procesos de ejemplificación y formas de realización en la sección Resumen precedente.

60

[0037] Debe comprenderse que se puede utilizar cualquier estructura de datos 12, lineal o no lineal, para almacenar los identificadores y resultados. Sin embargo, por cuestiones de velocidad de tratamiento, puede ser preferible usar una estructura de datos 12 con un sistema de índice eficaz, como una lista ordenada, una tabla hash o un árbol binario, como un árbol AVL.

65

Formas de realización específicas, aplicaciones y ejemplos

[0038] En lo sucesivo, se comentarán y ejemplificarán en detalle formas de realización de la invención, aplicaciones y ejemplos.

5

[0039] En algunas formas de realización de la invención, se usan cálculos y resultados previos en el procesamiento de las peticiones sucesivas para nuevos datos y cálculos. Con este fin, el proceso de extracción se diseña para almacenar en caché los resultados durante el procesamiento de las solicitudes de datos. Cuando se procesa una solicitud posterior, el proceso de extracción determina si ya se ha generado y almacenado en caché un resultado precedente apropiado. Si es así, el resultado precedente se usa en el procesamiento de la solicitud posterior. Ya que los cálculos previos no necesitan regenerarse, se puede reducir considerablemente el tiempo de procesamiento para la solicitud posterior.

10

[0040] En algunas formas de realización de la invención, se usan identificadores digitales (huellas digitales) para identificar la información almacenada en caché y, de esta manera, también se puede reutilizar un resultado almacenado en caché cuando se alcanza de forma diferente al cálculo precedente.

15

[0041] En algunas formas de realización de la invención, los identificadores digitales se almacenan en la caché. Concretamente, el identificador de la entrada para un procedimiento de cálculo se almacena junto con el identificador digital de la salida del procedimiento de cálculo. Por lo tanto, también se puede alcanzar el resultado final de una operación de muchas etapas cuando el/los resultado(s) intermedio(s) complejo(s) requerido(s) se ha(n) eliminado de la caché. Solo se necesita el identificador digital del/de los resultado(s) intermedio(s).

20

[0042] En algunas formas de realización de la invención, la caché se implementa mediante una estructura de datos que puede almacenar objetos heterogéneos, tales como tablas, subconjuntos de datos, matrices e identificadores digitales.

25

[0043] Las formas de realización de la invención pueden, por lo tanto, servir para minimizar, o al menos reducir, los tiempos de respuesta para un usuario que consulta un almacenamiento de datos mediante una consulta ejecutada recientemente por el mismo u otro usuario.

30

[0044] Las formas de realización de la invención también pueden servir para minimizar, o al menos reducir, el uso de memoria por la caché al reutilizar la misma entrada de caché para múltiples consultas o cálculos diferentes, en el caso de que dos consultas o cálculos produzcan el mismo resultado.

35

[0045] Las formas de realización de la invención se pueden aplicar para extraer cualquier tipo de información de cualquier tipo de base de datos conocida, como bases de datos relacionales, bases de datos postrelacionales, bases de datos orientadas a objetos, bases de datos jerárquicas, etc. Internet también se puede considerar como una base de datos en el contexto de la presente invención.

40

[0046] La Fig. 7 divulga una forma de realización específica de la invención, que consiste en un proceso de extracción o búsqueda de información que implica una consulta de base de datos con un cálculo del gráfico posterior basada en el resultado de la consulta. El resultado del cálculo del gráfico, denominado *Resultado del gráfico*, consiste habitualmente en datos que se agregan, clasifican o reagrupan en una, dos o múltiples dimensiones, por ejemplo en forma de un cubo multidimensional tal y como se ha comentado en la sección de Contexto.

45

[0047] En un primer paso, se define el *Campo* de búsqueda de información. En el caso de una consulta de base de datos, el campo se define mediante las tablas incluidas en una instrucción SELECCIONAR (o equivalente) y cómo estas se unen. Para una búsqueda en internet, el campo puede ser un índice de las páginas web encontradas, normalmente también organizadas como una o más tablas. De esta forma, la salida de la primera etapa es un conjunto de datos (cf. R0 en las Figs. 1-6).

50

[0048] En una segunda etapa, un usuario hace una *Selección* en el conjunto de datos, lo que provoca que un *Motor de inferencia* evalúe un número de filtros en el conjunto de datos. El motor de inferencia podría ser, por ejemplo, un motor de base de datos, una herramienta de consulta o una herramienta de inteligencia empresarial. Por ejemplo, en una consulta en una base de datos que almacena datos de órdenes transmitidas, esto podría requerir que el año de la orden sea "2007" y el grupo de productos "Productos lácteos". De esta forma, la selección puede definirse únicamente mediante una lista de campos incluidos y, para cada campo, una lista de valores seleccionados o, de manera más general, una condición.

55

60

[0049] Basándose en la selección (cf. S1 en las Figs. 1-6), el motor de inferencia ejecuta un procedimiento de cálculo (cf. P1 en las Figs. 1-6) para generar un *Subconjunto de datos* (cf. R1 en las Figs. 1-6) que representa una parte del campo (cf. R0 en las Figs. 1-6). El subconjunto de datos puede contener así un conjunto de registros de datos pertinentes del campo, o una lista de referencias (por ejemplo índices, indicadores o números binarios) a estos registros de datos pertinentes. En el ejemplo anterior, los registros de datos pertinentes

65

únicamente serían los registros de datos que pertenecen al año "2007" y al grupo de productos "Productos lácteos".

5 [0050] Si la selección no se ha hecho previamente, el motor de inferencia de la Fig. 7 se acciona para que calcule el subconjunto de datos. Sin embargo, si el cálculo se ha hecho previamente, el motor de inferencia se acciona en su lugar para que reutilice el resultado precedente accediendo a una estructura de datos específica: una "caché".

10 [0051] Con frecuencia, la etapa siguiente consiste en hacer algunos cálculos adicionales, por ejemplo agregación/agregaciones y/o ordenación/ordenaciones y/o agrupamiento(s), basados en el subconjunto de datos. En el ejemplo de la Figura 7, estos cálculos posteriores se llevan a cabo mediante un *Motor del gráfico* que calcula el *Resultado del gráfico* basado en el subconjunto de datos y un conjunto seleccionado de *Propiedades del gráfico* (cf. S2 en las Figs. 1-6). De esta forma, el motor del gráfico ejecuta un procedimiento de cálculo del gráfico (cf. P2 en las Figs. 1-6) para que genere el resultado del gráfico (cf. R2 en las Figs. 1-6). Si estos cálculos no se han hecho antes, el motor del gráfico de la Fig. 7 se acciona para que genere el resultado del gráfico. Sin embargo, si estos cálculos se han hecho antes, el motor del gráfico se acciona en su lugar para que reutilice el resultado precedente accediendo a la caché antes mencionada. El resultado del gráfico puede, a continuación, ser visualizado por un usuario en tablas dinámicas o, de forma gráfica, en gráficos 2D y 3D.

20 [0052] La Fig. 7 también ilustra el proceso de uso de la caché, donde f representa el algoritmo de hash que se opera para generar los identificadores digitales, donde ID1-ID4 representan los identificadores digitales generados de esta forma y las flechas de línea continua representan el flujo de datos para generar los identificadores ID1-ID4. Más adelante en la Fig. 7, las flechas discontinuas representan las búsquedas caché.

25 [0053] En la Fig. 7, cuando un usuario lleva a cabo una selección nueva, el motor de inferencia calcula el subconjunto de datos. Asimismo, el identificador ID1 para la selección junto con el campo se genera basándose en los filtros en la selección y el campo. Posteriormente, el identificador ID2 para el subconjunto de datos se genera a partir de la definición del subconjunto de datos, habitualmente una secuencia de bits que define el contenido del subconjunto de datos. Finalmente, el ID2 se almacena en la caché utilizando ID1 como identificador de búsqueda. Asimismo, la definición de subconjunto de datos se almacena en la caché usando ID2 como identificador de consulta.

35 [0054] En la Fig. 7, el cálculo del gráfico se desarrolla de manera similar. En este caso, hay dos conjuntos de información: el subconjunto de datos y las propiedades del gráfico pertinentes. El último es, normalmente aunque no restringido a ello, una función matemática junto con variables de cálculo y variables de clasificación (dimensiones). Ambos conjuntos de información se utilizan para calcular el resultado del gráfico y ambos conjuntos de información se utilizan también para generar el identificador ID3 para la entrada en el cálculo del gráfico. ID2 ya se había generado en la etapa precedente, e ID3 se genera como la primera etapa en el procedimiento de cálculo del gráfico.

40 [0055] El identificador ID3 se forma a partir de ID2 y las propiedades del gráfico pertinentes. ID3 puede verse como un identificador para una instancia específica de generación del gráfico, que incluye toda la información necesaria para calcular un resultado del gráfico específico. Además, se crea un identificador de resultado del gráfico ID4 a partir de la definición de resultado del gráfico, generalmente una secuencia de bits que define el resultado del gráfico. Finalmente, ID4 se almacena en la caché usando ID3 como identificador de búsqueda. Asimismo, la definición de resultado del gráfico se almacena en la caché usando ID4 como identificador de búsqueda.

50 [0056] En este ejemplo específico, se lleva a cabo un almacenamiento en caché del resultado en dos etapas, tanto en el procedimiento de inferencia como en el procedimiento de cálculo del gráfico. En el procedimiento de inferencia, ID1 e ID2 representan cosas diferentes: la selección y la definición del subconjunto de datos, respectivamente. Si dos selecciones diferentes producen el mismo subconjunto de datos, lo cual es bastante probable, el almacenamiento en caché en dos etapas (ID1:ID2; ID2: subconjunto de datos) provoca que el subconjunto de datos se almacene en caché solo una vez. Esto se denominará en lo sucesivo Combinación de objetos, es decir, diferentes objetos de datos que en la caché comparten la misma entrada caché. De forma similar, en el procedimiento de cálculo del gráfico, ID3 e ID4 representan cosas diferentes: la instancia de generación del gráfico y la definición del resultado del gráfico, respectivamente. Si dos instancias de generación del gráfico diferentes producen el mismo resultado del gráfico, lo cual es bastante probable, el almacenamiento en caché en dos etapas (ID3:ID4; ID4: resultado del gráfico) provoca que el resultado del gráfico se almacene en caché solo una vez.

65 [0057] Además, almacenando en caché ID3, el resultado del gráfico también se puede recrear si la definición del subconjunto de datos se ha eliminado de la caché. Esto supone una ventaja relevante, ya que la definición del subconjunto de datos puede ser muy amplia y, por lo tanto, propensa a ser eliminada de la caché si se implementa un mecanismo de eliminación de la caché. Más adelante, se describe un ejemplo no limitativo de dicho mecanismo.

[0058] Durante el proceso de extracción, los identificadores se calculan a partir de la selección, las propiedades del gráfico pertinentes, etc. y se usa para buscar resultados de cálculo posiblemente almacenados en caché, como indican las flechas discontinuas en la Fig. 7. Si se encuentra el identificador, se reutilizará el resultado almacenado en caché correspondiente. Si no se encuentra, el proceso de extracción generará nuevos identificadores y los almacenará en caché con el resultado respectivo.

[0059] Para ejemplificar de forma adicional el proceso de extracción, cabe considerar la selección antes mencionada del año de orden "2007" y del grupo de productos "Productos lácteos". La primera etapa consiste en generar un identificador digital ID1 en función de esta selección, por ejemplo (escrito en notación hexadecimal): "31dca7ad013964891df428095ad9b78ad7a69eaaa1ca3886bcf05d8f8184e84a".

[0060] Con el objetivo de mantener la brevedad, en el siguiente ejemplo, cada identificador se representa por sus 4 caracteres iniciales. Así, ID1 se vuelve, en cambio, "31dc". Asimismo, por motivos de claridad, las tablas ilustrativas siguientes incluyen etiquetas de identificador, por ejemplo "ID1:" delante de los identificadores digitales. Esto no es necesario en la solución real.

[0061] El proceso de extracción posterior es el siguiente: cuando se ha generado ID1, este se busca en la caché. La primera vez que se lleva a cabo la selección, este identificador no se encontrará en la caché, por lo que el subconjunto de datos resultante debe calcularse de la forma normal. Una vez esto se haya hecho, se puede generar ID2 desde el subconjunto de datos para que sea, por ejemplo, "d2b8". A continuación, ID1 se almacena en caché, señalando a ID2; e ID2 se almacena, señalando a la secuencia de bits que define el subconjunto de datos resultante. Esta secuencia de bits puede tener un tamaño considerable. El contenido de la caché se muestra en Tabla 1 más abajo.

Tabla 1:

ID	Valor en caché
ID1:31dc	ID2:d2b8
ID2:d2b8	<registros de datos en el subconjunto de datos resultante>

[0062] En la siguiente ocasión en la que se realice la misma selección, el proceso será diferente: ahora, ID1 se encuentra en la caché, señalando a "ID2:d2b8", que a su vez se usa para una segunda búsqueda, tras lo cual la secuencia de bits del subconjunto de datos resultante se encuentra, recupera y utiliza en lugar de un cálculo, que consumiría mucho tiempo.

[0063] Cabe considerar ahora el caso en el que se realiza una selección diferente, pero que produce el mismo subconjunto de datos resultante. Por ejemplo, puede ocurrir que un usuario seleccione exactamente los clientes que han comprado "Productos lácteos" sin solicitar explícitamente "Productos lácteos" y estos han comprado exclusivamente productos lácteos. En este caso, ID1 se genera como, por ejemplo, "f142" y no se encontrará en la caché. Así, el subconjunto de datos resultante debe calcularse de la forma normal. Una vez esto esté hecho, ID2 se puede generar a partir del subconjunto de datos y se encuentra como "d2b8", que ya se ha almacenado en la caché. Así, la necesidad del algoritmo solo añade una entrada a la caché, donde "ID1:f142" señala a "ID2:d2b8". El contenido de la caché se muestra en Tabla 2 más abajo.

Tabla 2:

ID	Valor en caché
ID1:f142	ID2:d2b8
ID1:31dc	ID2:d2b8
ID2:d2b8	<registros de datos en el subconjunto de datos resultante>

[0064] Esta vez no se ha ahorrado tiempo de cálculo, pero las entradas caché se reutilizan para prevenir que la caché aumente de forma innecesaria. Así, tanto "ID1:f142" como "ID1:31dc" señalan a la entrada caché que contiene el mismo subconjunto de datos resultante: "ID2:d2b8", y ambos se pueden usar en búsquedas posteriores. Esto es, por lo tanto, un ejemplo de la "combinación de objetos" antes mencionada.

[0065] Una ventaja adicional de almacenar en caché los identificadores digitales resultará evidente cuando se lleve a cabo el cálculo del gráfico posterior. Así, se asume que las selecciones anteriores se han llevado a cabo y se ha realizado el cálculo del gráfico posterior. ID3 e ID4 se han generado como "e40A" y "7505", respectivamente, y almacenado en la caché. El contenido de la caché se muestra en la Tabla 3 más abajo.

Tabla 3:

ID	Valor en caché
ID1:f142	ID2:d2b8
ID1:31dc	ID2:d2b8
ID2:d2b8	<registros de datos en el subconjunto de datos resultante>
ID3:e40A	ID4:7505
ID4:7505	<matriz de números que representa el resultado del gráfico>

5 [0066] De las cinco entradas de la Tabla 3, es muy probable que una sea considerablemente mayor que las demás: "ID2:d2b8", que contiene toda la secuencia de bits que define el subconjunto de datos potencialmente grande. Su tamaño la hace una candidata para ser eliminada cuando/si la caché se mantiene, tal y como se describe de forma adicional más adelante. Así, después de un tiempo, el contenido de la caché puede ser tal y como se muestra en la Tabla 4 más adelante.

Tabla 4:

ID	Valor en caché
ID1:f142	ID2:d2b8
ID1:31dc	ID2:d2b8
ID3:e40A	ID4:7505
ID4:7505	<matriz de números que representa el resultado del gráfico>

10 [0067] Sin embargo, ya que los identificadores digitales se han almacenado en caché, todavía es posible obtener el resultado del gráfico sin tener que recalcular el subconjunto de datos intermedio. En cambio, cuando se lleva a cabo la selección, se calcula ID1. A continuación, se realiza una búsqueda de ID1 en la caché, que provoca que se recupere ID2. Posteriormente, ID3 se genera a partir de la combinación de las propiedades del gráfico pertinente e ID2. Se lleva a cabo una búsqueda de ID3 en la caché y se recupera ID4. Finalmente, se realiza una búsqueda de ID4 en la caché y se recupera el resultado del gráfico. Por lo tanto, el resultado del gráfico se encuentra sin cálculos pesados, sino simplemente basándose en los identificadores digitales, que se pueden generar mediante operaciones rápidas y eficientes durante el procesamiento.

20 [0068] A partir de lo anterior, se entiende que los identificadores digitales deberían ser únicos de modo que el significado de cada identificador en la caché no sea ambiguo. En una forma de realización, los identificadores digitales se generan utilizando un algoritmo o función hash. Los algoritmos de hash son transformaciones que se sirven de una entrada de tamaño arbitrario (el mensaje) y devuelven una cadena de tamaño fijo llamada valor de hash (resumen de mensaje). Normalmente el algoritmo corta y mezcla, por ejemplo, sustituye o traspone, la entrada para crear una huella digital de la misma. Los algoritmos de hash más sencillos y antiguos son operaciones módulo por primo sencillas. Los algoritmos de hash se usan para una variedad de fines computacionales, incluyendo la criptografía. En términos generales, un algoritmo de hash debería comportarse, dentro de lo posible, como una función aleatoria, generando cualquier cadena de tamaño fijo posible con igual "probabilidad", mientras todavía es verdaderamente determinista.

30 [0069] Hay múltiples algoritmos de hash conocidos y usados frecuentemente que se pueden utilizar para generar los identificadores digitales anteriormente mencionados. Los diferentes algoritmos de hash se optimizan para usos diferentes, donde algunos se optimizan para una computación eficaz y rápida del valor de hash, mientras que otros se diseñan para una seguridad criptográfica alta. Un algoritmo con una alta seguridad criptográfica se diseña para que sea difícil calcular un mensaje que coincida con un valor de hash determinado en un tiempo razonable, y para encontrar un segundo mensaje que genere el mismo valor de hash que un primer mensaje determinado. Dichos algoritmos de hash incluyen SHA (por sus siglas en inglés, algoritmo de hash seguro) y MD5 (por sus siglas en inglés, algoritmo de resumen de mensaje 5). Los algoritmos de hash eficientes en el procesamiento habitualmente muestran una menor seguridad criptográfica. Dichos algoritmos de hash incluyen los algoritmos FNV (Fowler/Noll/Vo), diseñados para que sean rápidos y mantener generalmente un índice de colisión muy bajo. Un algoritmo FNV habitualmente comienza con una base de desplazamiento, que en principio podría ser cualquier cadena de valores aleatoria, pero generalmente, por tradición, es siempre la firma del inventor en código hexadecimal a través del algoritmo FNV-0 original. Para generar un valor de hash FNV de 256-bits, normalmente se usa la base de desplazamiento siguiente:
 "0xdd268dbcaac550362d98c384c4e576ccc8b1536847b6bbb31023b4c8caee0535".

45 [0070] Para cada byte en la entrada al algoritmo de hash, el desplazamiento se multiplica primero por un número primo grande, posteriormente se compara con el byte de la entrada y finalmente se calcula la diferencia simétrica a nivel de bits (XOR) para formar el valor de hash para el bucle siguiente. Los números primos apropiados se

encuentran en la literatura disponible. Cualquier número primo grande funcionará, pero algunos son más resistentes a la colisión que otros.

5 [0071] Los identificadores digitales se pueden generar utilizando cualquier algoritmo de hash que sea razonablemente resistente a la colisión. En una forma de realización, los identificadores se generan utilizando un algoritmo de hash rápido con alta resistencia a la colisión y baja seguridad criptográfica.

10 [0072] En una forma de realización específica, se puede crear un identificador de 256-bits mediante concatenación de cuatro valores de hash FVN de 64-bits, donde cada uno se ha generado utilizando un multiplicador primo diferente. Usando cuatro valores de hash más cortos y concatenándolos, el identificador se puede generar más rápido. Para aumentar la velocidad de la generación del identificador, el algoritmo se puede modificar para que use no solo un byte de la entrada por bucle, sino cuatro bytes. Este puede suponer una pérdida de seguridad criptográfica, mientras que la resistencia a la colisión se mantiene aproximadamente igual.

15 [0073] Los identificadores con una longitud de al menos 256 bits pueden producir una resistencia a la colisión beneficiosa. Un valor de hash de 256-bits significa que hay aproximadamente $1E+77$ valores identificadores posibles. Este número se puede comparar con el número de átomos en el universo, que se ha estimado en $1E+80$. Esto significa que el riesgo de colisiones, es decir, el riesgo de que dos selecciones/subconjuntos de datos/propiedades del gráfico/resultados del gráfico diferentes produzcan el mismo identificador no es solo extremadamente pequeño, sino insignificante. De esta forma, se puede decir con certeza que el riesgo de colisiones es aceptablemente pequeño. Esto significa que, aunque el algoritmo de hash no genera identificadores teóricamente únicos, sí que genera, sin embargo, identificadores estadísticamente únicos. No obstante, debe entenderse que los identificadores con menores longitudes de bit, como 64 o 128 bits, pueden ser lo bastante estadísticamente únicos para una aplicación específica.

25 [0074] Como se ha mencionado anteriormente, se puede implementar un mecanismo de eliminación para eliminar la caché de entradas antiguas o sin usar. Una estrategia puede ser eliminar la(s) entrada/entradas con menos uso en la caché. Sin embargo, se puede implementar un mecanismo de eliminación más avanzado para apoyar la optimización tanto del uso del procesador como del uso de la memoria. Una forma de realización de dicho mecanismo de eliminación avanzado opera en tres parámetros: Uso, Tiempo de cálculo y Memoria necesaria.

35 [0075] El parámetro de Uso es un valor numérico que puede considerar tanto si se ha accedido a una entrada "recientemente, pero no a menudo" como si se ha accedido a la entrada "a menudo, pero no recientemente". Esto se puede lograr asociando cada entrada a un parámetro de uso U que se aumenta mediante, por ejemplo, una unidad cada vez que se accede a la entrada, pero reduce su valor exponencialmente, o por cualquier otra función, a lo largo del tiempo. En una implementación, todos los valores de U en la caché se reducen periódicamente en una cantidad fija. Así, el parámetro de uso tiene una vida media, similar a la desintegración radiactiva. El valor de U reflejará ahora con qué frecuencia y hace cuánto tiempo se ha accedido a la entrada.

40 [0076] Si el tiempo de procesador necesario para calcular una entrada es considerable, entonces la entrada se debería mantener durante más tiempo en la caché. Por el contrario, si el tiempo de procesador necesario para el cálculo es poco, entonces el coste de recalcularse es bajo y el beneficio de mantener la entrada en la caché también es bajo. Así, cada entrada se asocia a un parámetro temporal T que representa el tiempo de cálculo estimado.

45 [0077] Si el espacio de memoria necesario para memorizar una entrada es considerable, entonces supone el uso de muchos de los recursos de la caché para mantenerlo y debería eliminarse de la caché antes que una entrada que requiera menos espacio de memoria. Por el contrario, una entrada que requiera poco espacio de memoria se puede mantener durante más tiempo en la caché. Así, cada entrada se asocia con un parámetro de memoria M que representa la memoria necesaria estimada.

50 [0078] Por cada entrada en la caché, los valores de los parámetros U, T y M se evalúan mediante una función de peso W obtenida por: $W = U * T / M$.

55 [0079] Un valor de W alto para una entrada indica que hay buenas razones para mantener esta entrada en la caché. Así, las entradas con valores W altos deberían mantenerse en la caché y aquellas con valores W bajos deberían ser eliminadas.

60 [0080] Un mecanismo de eliminación eficaz puede implicar ordenar la caché según los valores W y eliminar la caché ordenada en un extremo, es decir, las entradas con los valores W más bajos. Un método posible, pero no necesario, para mantener una caché ordenada podría ser almacenar los identificadores, resultados y valores U, T, M y W como un árbol AVL (Adelson-Velsky y Landis), es decir, un árbol binario de búsqueda equilibrado.

65 [0081] El mecanismo de eliminación puede eliminar intermitentemente todas las entradas con un valor W que estén por debajo de un valor umbral predeterminado.

5 [0082] De forma alternativa, el mecanismo de eliminación se puede controlar mediante la cantidad de memoria disponible en el ordenador o la proporción de memoria disponible de la memoria total. Así, siempre que el tamaño de la memoria caché campo un valor umbral de memoria, el mecanismo de eliminación elimina entradas de las entradas caché basándose en sus respectivos valores W. Ajustando el umbral de memoria, es posible adaptar el tamaño de caché a las condiciones del hardware local, por ejemplo intercambiando la potencia de procesamiento con la memoria. Por ejemplo, es posible compensar un procesador más lento en un ordenador añadiendo más memoria principal al ordenador y aumentando el umbral de memoria. Así, se retendrán más resultados en la caché y la necesidad de procesamiento se reducirá.

10 [0083] Las formas de realización de la invención también se refieren a un aparato para realizar cualquiera de los algoritmos, métodos, procesos y procedimientos descritos previamente. Este aparato puede construirse especialmente para el fin requerido o puede incluir un ordenador general que se activa o reconfigura selectivamente mediante un programa informático almacenado en el ordenador.

15 [0084] La Fig. 8 es un diagrama de bloques de un entorno de ordenador para implementar cualquiera de las formas de realización de la invención. Un usuario 1 interactúa con un sistema de tratamiento de datos 2, que incluye un procesador 3 que ejecuta el software del sistema operativo, así como uno o más programas de aplicación que implementan una forma de realización de la invención. El usuario introduce información en el sistema de tratamiento de datos 2 usando uno o más dispositivos de entrada conocidos 4, como un ratón, un teclado, un panel táctil, etc. De forma alternativa, la información se puede introducir con o sin intervención del usuario con cualquier otro tipo de dispositivo de entrada, tal como un lector de tarjetas, un lector óptico u otro sistema informático. La respuesta visual se puede hacer llegar al usuario mostrando caracteres, símbolos gráficos, ventanas, teclas, etc., en una pantalla 5. El sistema de tratamiento de datos incluye además la memoria 10 antes mencionada. El software ejecutado por el procesador 3 almacena información acerca de su operación en la memoria 10 y recupera información apropiada desde la memoria 10. La memoria 10 incluye habitualmente una memoria principal (como RAM, memoria caché, etc.) y una memoria secundaria no volátil (disco duro, memoria flash, soporte extraíble). La base de datos se puede almacenar en la memoria 10 del sistema de tratamiento de datos o se puede acceder a ella en un dispositivo de memoria externa a través de una interfaz de comunicaciones 6 en el sistema de tratamiento de datos 2.

25 [0085] La invención se ha descrito antes principalmente en referencia a algunas formas de realización. Sin embargo, como el experto en la técnica podrá notar fácilmente, también son posibles otras formas de realización aparte de las que se han descrito previamente.

35 [0086] Por ejemplo, la presente invención no solo se puede aplicar para calcular cubos multidimensionales, sino que puede ser útil en cualquier situación en la que se extrae información de una base de datos que se sirve de una cadena de cálculos.

40 [0087] Además, el proceso de extracción inventivo se puede aplicar a una cadena de cálculos que implica más de dos cálculos consecutivos. Por ejemplo, cada uno de dos o más resultados intermedios en una cadena de cálculos se puede almacenar en caché y recuperar posteriormente de forma similar al resultado intermedio descrito anteriormente.

45 [0088] Asimismo, el proceso de extracción inventivo no requiere almacenar en caché y posteriormente recuperar el resultado final, sino que puede operar solo para almacenar en caché y recuperar uno o más resultados intermedios en una cadena de cálculos.

50 [0089] Adicionalmente, se debe tener en cuenta que la etapa inicial de extracción de un conjunto o campo de datos inicial de la base de datos se puede omitir y el proceso de extracción puede, en cambio, operar directamente en la base de datos.

REIVINDICACIONES

1. Método implementado en el ordenador para extraer información a partir de una base de datos, donde dicho método incluye una cadena secuencial de cálculos principales que incluye un primer cálculo principal (P1) que opera un primer elemento de selección (S1) en un conjunto de datos (R0) que representa la base de datos para producir un resultado intermedio (R1), y un segundo cálculo principal (P2) que opera un segundo elemento de selección (S2) en el resultado intermedio (R1) para producir un resultado final (R2), donde dicho método incluye además la recuperación del resultado final mediante las etapas de:

- (a) calcular un primer valor de identificador de selección (ID1) como una huella digital estadísticamente única generada por una función hash de al menos el primer elemento de selección (S1);
 (b) buscar, en los objetos de la estructura de datos, el primer valor de identificador de selección (ID1) y, si se encuentra el primer valor de identificador de selección (ID1), localizar y recuperar un primer identificador de resultado (ID2), almacenado con el primer valor de identificador de selección (ID1), como objetos asociados en una iteración precedente;
 (c) si el primer identificador de resultado (ID2) se encuentra en la subetapa (b),

calcular un segundo valor de identificador de selección (ID3) como una huella digital estadísticamente única generada por una función hash de al menos el segundo elemento de selección (S2) y el primer identificador de resultado (ID2) recuperado, y
 buscar, en los objetos de la estructura de datos, el segundo valor de identificador de selección (ID3) y, si se encuentra el segundo valor de identificador de selección (ID3), localizar y recuperar un resultado final (R2), almacenado con el segundo valor de identificador de selección (ID3), como objetos asociados en una iteración precedente;

- (d) si el primer identificador de resultado (ID2) no se encuentra en la subetapa (b),

ejecutar el primer cálculo principal (P1) para producir el resultado intermedio (R1) y el primer valor de identificador de resultado (ID2) como una huella digital generada por una función hash del resultado intermedio (R1),
 almacenar el primer valor de identificador de selección (ID1) y el primer valor de identificador de resultado (ID2) como objetos asociados en la estructura de datos; y
 almacenar el primer valor de identificador de resultado (ID2) y el resultado intermedio (R1) como objetos asociados en la estructura de datos,
 calcular un segundo valor de identificador de selección (ID3) como una huella digital estadísticamente única generada por una función hash del primer valor de identificador de resultado (ID2) y el segundo elemento de selección (S2), y
 buscar en los objetos de la estructura de datos basándose en el segundo valor de identificador de selección (ID3) y, si se encuentra el segundo valor de identificador de selección (ID3), localizar y recuperar un resultado final (R2) almacenado con el segundo valor de identificador de selección (ID3) como objetos asociados en una iteración precedente;

- (e) si el resultado final (R2) no se encuentra en la subetapa (c) o (d),
 buscar, en los objetos de la estructura de datos basados en el primer valor de identificador de resultado (ID2);
 (f) si el primer valor de identificador de resultado (ID2) no se encuentra en la subetapa (e),

ejecutar el primer cálculo principal (P1) para producir el resultado intermedio (R1) y el primer valor de identificador de resultado (ID2) como una huella digital generada por una función hash del resultado intermedio (R1),
 almacenar el primer valor de identificador de resultado (ID2) y el resultado intermedio (R1) como objetos asociados en la estructura de datos, y
 ejecutar el segundo cálculo principal (P2) para producir el resultado final (R2) y almacenar el segundo valor de identificador de selección (ID3) y el resultado final (R2) como objetos asociados en la estructura de datos; y

- (g) si el primer valor de identificador de resultado (ID2) se encuentra en la subetapa (e),

recuperar el resultado intermedio (R1) almacenado con el primer valor de identificador de resultado (ID2) como objetos asociados en una iteración precedente, y
 ejecutar el segundo cálculo principal (P2) para producir el resultado final (R2) y almacenar el segundo valor de identificador de selección (ID3) y el resultado final (R2) como objetos asociados en la estructura de datos.

2. Método según la reivindicación 1, donde la huella digital comprende al menos 256 bits.

3. Método según la reivindicación 1 o la reivindicación 2, que incluye además la etapa de eliminar selectivamente los registros de datos que contienen dichos objetos asociados en la estructura de datos, basándose al menos en el tamaño del registro de datos.
- 5 4. Método según la reivindicación 3, donde la etapa de eliminación selectiva se configura para que provoque la eliminación de los registros de datos que contienen dicho primer resultado (R1).
- 10 5. Método según la reivindicación 3 o 4, que comprende además la etapa de asociar cada registro de datos con un valor de peso, que se calcula en función de un parámetro de uso para cada registro de datos, un parámetro de cálculo de tiempo para cada registro de datos y un parámetro de tamaño para cada registro de datos, donde el parámetro de uso es un valor numérico que representa con qué frecuencia y hace cuánto tiempo se ha accedido al registro de datos, donde el parámetro de cálculo de tiempo representa el tiempo de cálculo estimado para el registro de datos y el parámetro de tamaño representa el tamaño del registro de datos.
- 15 6. Método según la reivindicación 5, donde el valor de peso se calcula evaluando una función de peso obtenida por $W=U*T/M$, donde U es el parámetro de uso, T es el parámetro de tiempo de cálculo y M es el parámetro de tamaño.
- 20 7. Método según la reivindicación 5 o 6, donde el valor del parámetro de uso se incrementa siempre que se accede al registro de datos, mientras que disminuye exponencialmente en función del tiempo.
8. Método según cualquiera de las reivindicaciones 4-7, donde la etapa de eliminación selectiva se basa en el valor de peso del registro de datos en la estructura de datos.
- 25 9. Método según cualquiera de las reivindicaciones 4-8, donde la etapa de eliminación selectiva se desencadena basándose en una comparación entre un tamaño actual de la estructura de datos y un valor umbral.
- 30 10. Método según cualquiera de las reivindicaciones precedentes, donde la base de datos es una base de datos dinámica y, por lo tanto, una base de datos que puede cambiar en cualquier momento, y el primer valor de identificador de selección (ID1) se calcula en función de al menos el primer elemento de selección (S1) y el conjunto de datos (R0).
- 35 11. Método según cualquiera de las reivindicaciones precedentes, donde el primer elemento de selección (S1) define un conjunto de campos en el conjunto de datos (R0) y una condición para cada campo, donde el resultado intermedio (R1) es representativo de un subconjunto del conjunto de datos (R0), donde el segundo elemento de selección (S2) define una función matemática, una o más variables de cálculo incluidas en el resultado intermedio (R1) y una o más variables de clasificación incluidas en el resultado intermedio (R1), y donde el resultado final (R2) es una estructura de datos de cubo multidimensional que contiene el resultado de operar la función matemática en dichas una o más variables de cálculo para cada valor único de cada variable de clasificación.
- 40 12. Medio legible por ordenador que almacena un programa informático que, cuando se ejecuta por medio de un ordenador, es apto para llevar a cabo el método de cualquiera de las reivindicaciones 1-11.
- 45 13. Aparato para extraer información a partir de una base de datos, donde dicho aparato comprende medios para ejecutar una cadena secuencial de cálculos principales que comprende un primer cálculo principal (P1) que opera un primer elemento de selección (S1) en un conjunto de datos (R0) que representa la base de datos para producir un resultado intermedio (R1), y un segundo cálculo principal (P2) que opera un segundo elemento de selección (S2) en el resultado intermedio (R1) para producir un resultado final (R2), donde dicho aparato comprende además medios para recuperar el resultado final mediante las etapas de:
- 50 (a) calcular un primer valor de identificador de selección (ID1) como una huella digital estadísticamente única generada por una función hash de al menos el primer elemento de selección (S1);
- 55 (b) buscar, en los objetos de la estructura de datos, el primer valor de identificador de selección (ID1) y, si se encuentra el primer valor de identificador de selección (ID1), localizar y recuperar un primer identificador de resultado (ID2) almacenado con el primer valor de identificador de selección (ID1) como objetos asociados en una iteración precedente;
- (c) si el primer identificador de resultado (ID2) se encuentra en la subetapa (b),
- 60 calcular un segundo valor de identificador de selección (ID3) como una huella digital estadísticamente única generada por una función hash de al menos el segundo elemento de selección (S2) y el primer identificador de resultado (ID2) recuperado, y
- 65 buscar, en los objetos de la estructura de datos, el segundo valor de identificador de selección (ID3) y, si se encuentra el segundo valor de identificador de selección (ID3), localizar y recuperar un resultado final (R2) almacenado con el segundo valor de identificador de selección (ID3) como objetos asociados en una iteración precedente;

(d) si el primer identificador de resultado (ID2) no se encuentra en la subetapa (b),

5 ejecutar el primer cálculo principal (P1) para producir el resultado intermedio (R1) y el primer valor de
identificador de resultado (ID2) como una huella digital generada por una función hash del resultado
intermedio (R1),
almacenar el primer valor de identificador de selección (ID1) y el primer valor de identificador de resultado
(ID2) como objetos asociados en la estructura de datos; y
10 almacenar el primer valor de identificador de resultado (ID2) y el resultado intermedio (R1) como objetos
asociados en la estructura de datos,
calcular un segundo valor de identificador de selección (ID3) como una huella digital estadísticamente
única generada por una función hash del primer valor de identificador de resultado (ID2) y el segundo
elemento de selección (S2), y
15 buscar en los objetos de la estructura de datos basándose en el segundo valor de identificador de
selección (ID3) y, si se encuentra el segundo valor de identificador de selección (ID3), localizar y
recuperar un resultado final (R2) almacenado con el segundo valor de identificador de selección (ID3)
como objetos asociados en una iteración precedente;

(e) si el resultado final (R2) no se encuentra en la subetapa (c) o (d),
20 buscar en los objetos de la estructura de datos basándose en el primer valor de identificador de resultado
(ID2);

(f) si el primer valor de identificador de resultado (ID2) no se encuentra en la subetapa (e)

25 ejecutar el primer cálculo principal (P1) para producir el resultado intermedio (R1) y el primer valor de
identificador de resultado (ID2) como una huella digital generada por una función hash del resultado
intermedio (R1),
almacenar el primer valor de identificador de resultado (ID2) y el resultado intermedio (R1) como objetos
asociados en la estructura de datos, y
30 ejecutar el segundo cálculo principal (P2) para producir el resultado final (R2) y almacenar el segundo
valor de identificador de selección (ID3) y el resultado final (R2) como objetos asociados en la estructura
de datos; y

(g) si el primer valor de identificador de resultado (ID2) se encuentra en la subetapa (e)

35 recuperar el resultado intermedio (R1) almacenado con el primer valor de identificador de resultado (ID2)
como objetos asociados en una iteración precedente, y
ejecutar el segundo cálculo principal (P2) para producir el resultado final (R2) y almacenar el segundo
valor de identificador de selección (ID3) y el resultado final (R2) como objetos asociados en la estructura
de datos.

40

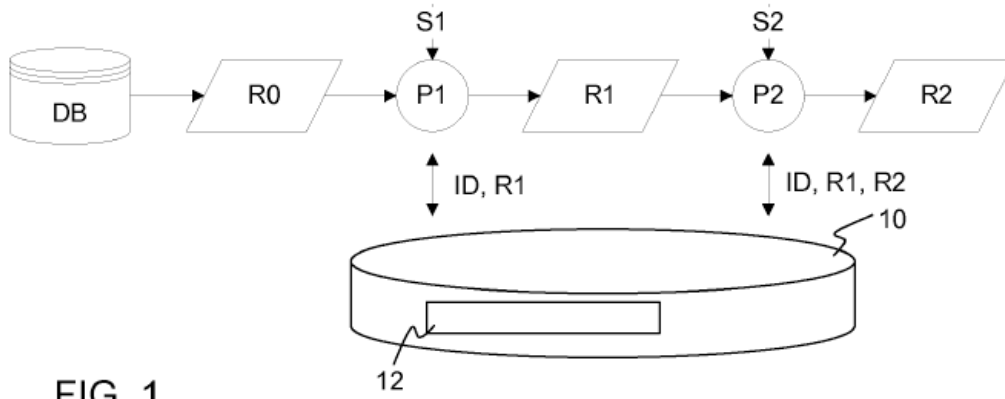


FIG. 1

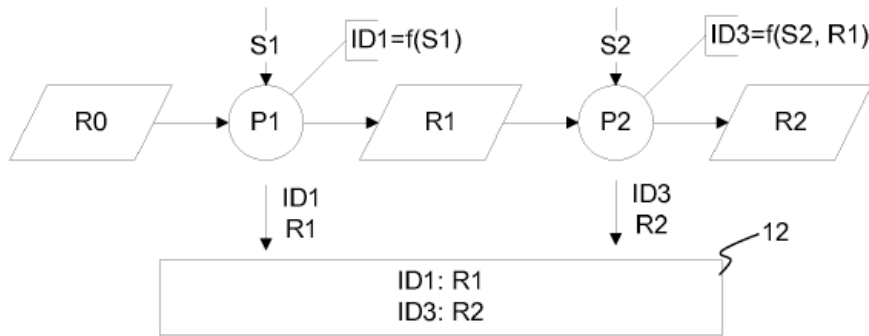


FIG. 2

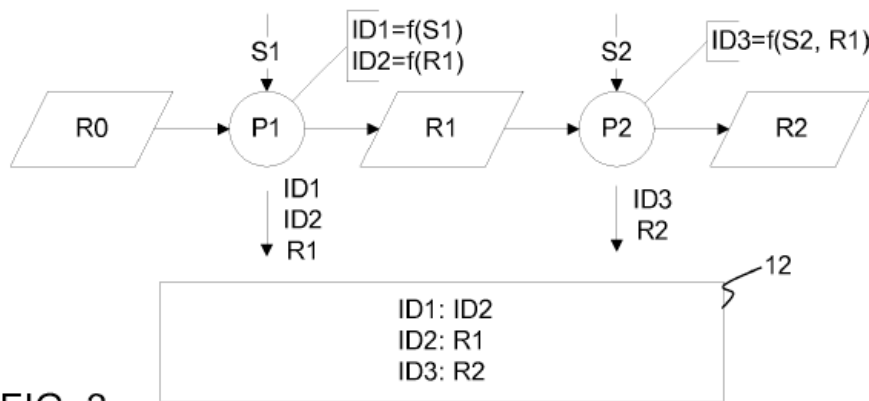


FIG. 3

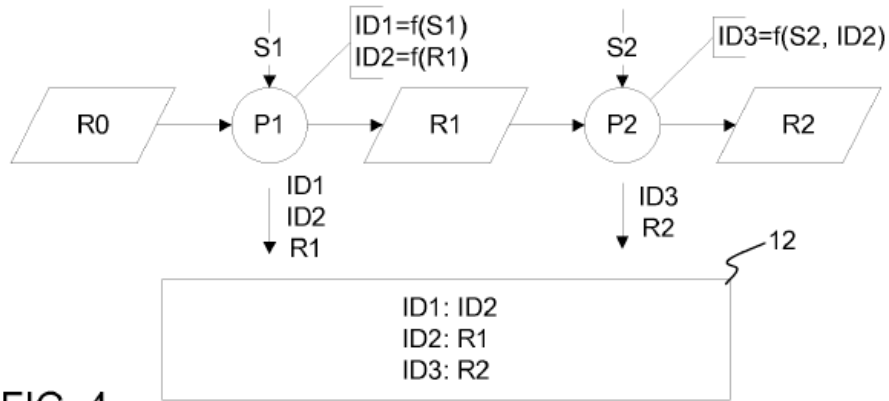


FIG. 4

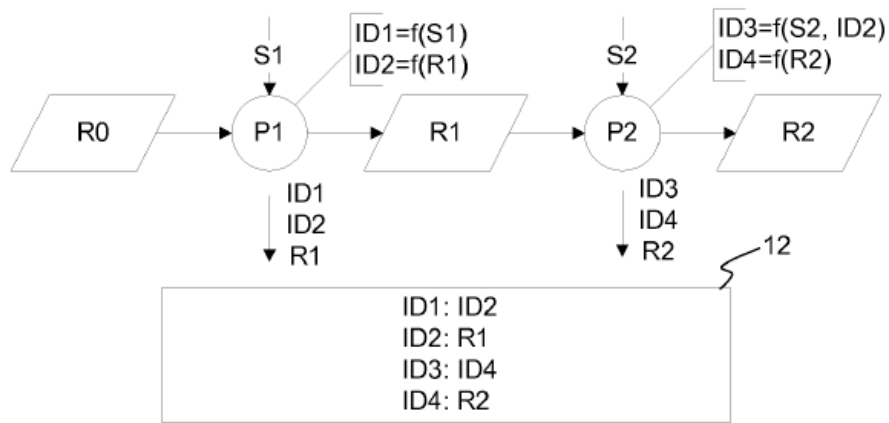


FIG. 5

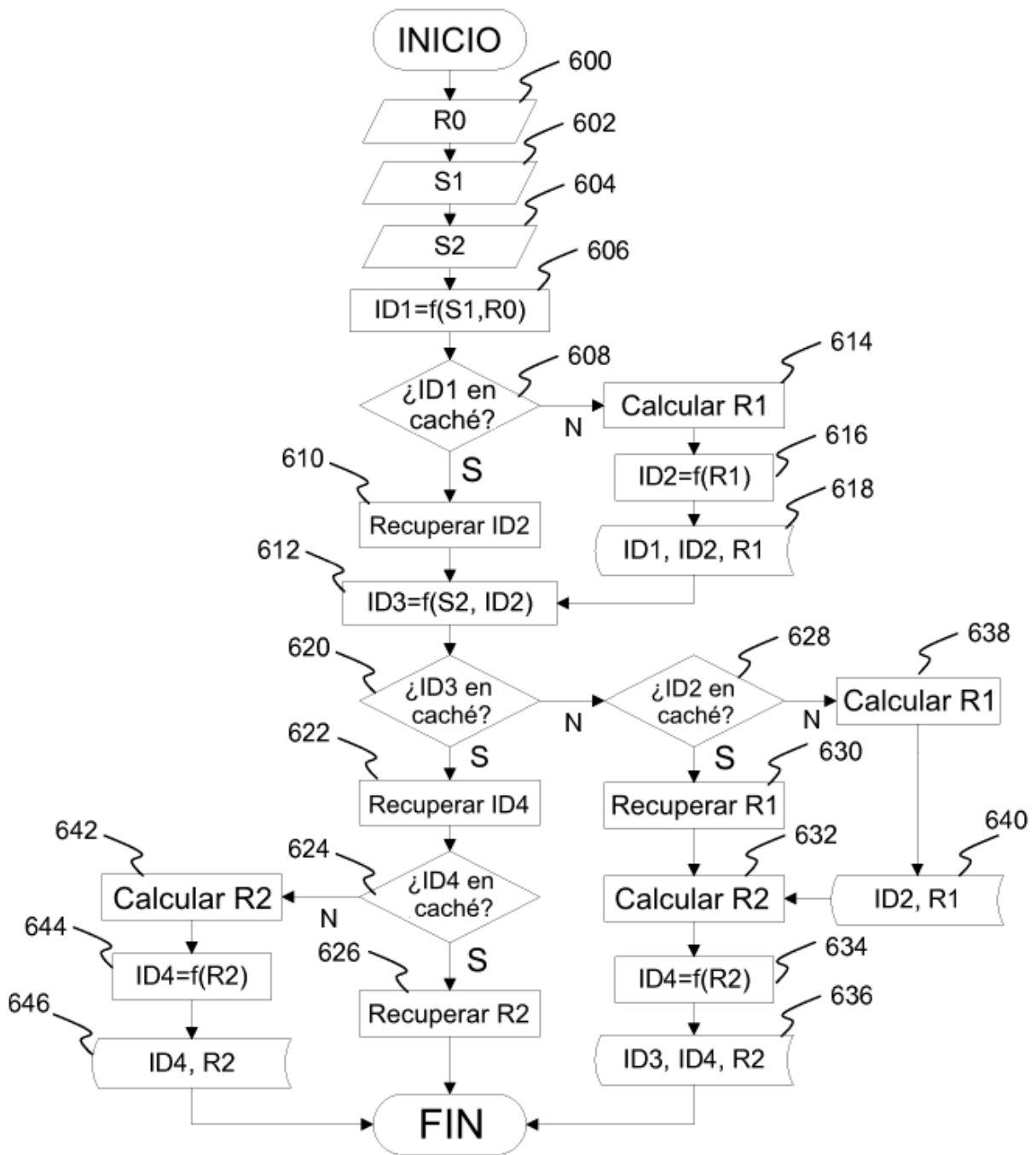


FIG. 6

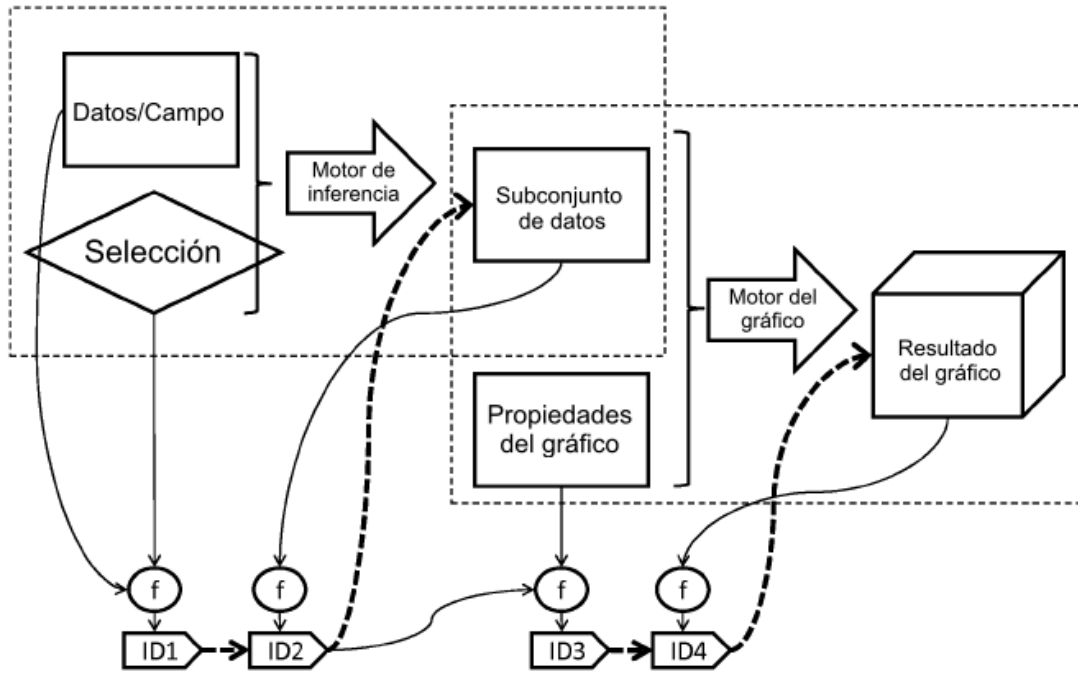


FIG. 7

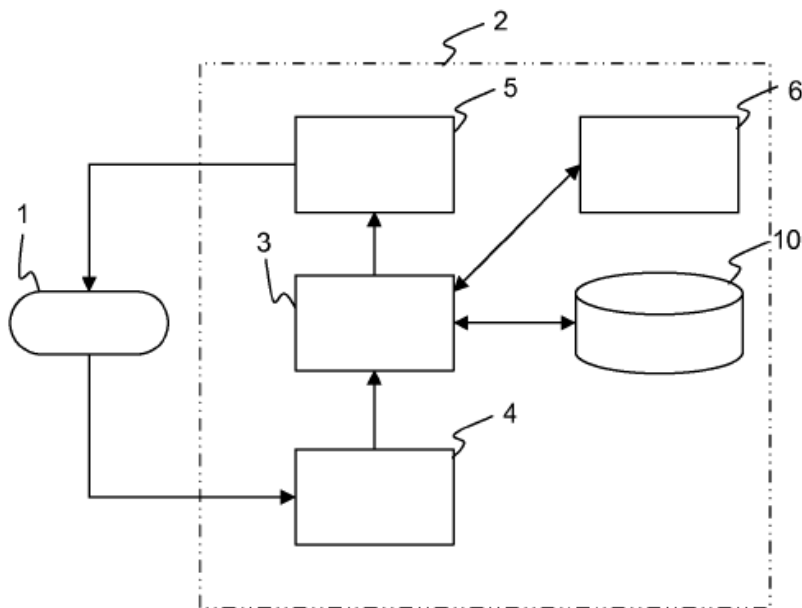


FIG. 8