(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 593 779**

(51) Int. CI.:

G06F 17/30 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **18.05.2004 PCT/US2004/015528**

(87) Fecha y número de publicación internacional: **02.12.2004 WO04104774**

(96) Fecha de presentación y número de la solicitud europea: **18.05.2004 E 04752528 (2)**

(97) Fecha y número de publicación de la concesión europea: **13.07.2016 EP 1629406**

(54) Título: **Limitar la exploración de relaciones poco ordenadas y/o agrupadas usando correspondencias casi ordenadas**

(30) Prioridad:

19.05.2003 US 471691 P

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
13.12.2016

(73) Titular/es:

HUAWEI TECHNOLOGIES CO., LTD. (100.0%)
Huawei Administration Building, Bantian
Longgang District, Shenzhen, Guangdong 518129, CN

(72) Inventor/es:

METZGER, JOHN, K.;
ZANE, BARRY, M. y
HINSHAW, FOSTER, D.

(74) Agente/Representante:

LEHMANN NOVO, María Isabel

DESCRIPCIÓN

Limitar la exploración de relaciones poco ordenadas y/o agrupadas usando correspondencias casi ordenadas

Solicitud relacionada

Esta solicitud reivindica el beneficio de la solicitud provisional estadounidense n.º 60/471.691, presentada el 19 de mayo de 2003. Las enseñanzas de la solicitud anterior se incorporan en el presente documento a modo de referencia.

Antecedentes de la invención

Los índices de bases de datos relacionales tienen normalmente una entrada para cada registro de datos en la relación objetivo. Cada entrada de estos índices contiene un valor de clave y un puntero al registro de datos. Estas entradas están ordenadas por el valor de clave. Características de tales índices incluyen mantener campos de datos adicionales en cada entrada y procedimientos de acceso a alta velocidad y de mantenimiento tales como "árboles b". Estas características son normalmente útiles en sistemas transaccionales en los el acceso a los registros se realiza de uno en uno. Sin embargo, su mantenimiento y exploración son complicados, especialmente cuando se busca un gran número de registros.

Documentos relevantes de la técnica incluyen '*Oracle9i, Data Warehousing Guide*, *Release 2 (9.2)*', marzo de 2002, n.º de referencia A96520-01, XP-002482800; la patente estadounidense n.º 6.014.656 a nombre de Hallmark et al.; '*Oracle8i Enterprise Edition Partitioning Option*', febrero de 1999, XP-002482801; '*Load Balancing: A case study of a pharmaceutical drug candidate database*', de Ben Miled et al., Actas del Tercer Simposio de IEEE sobre Bioinformática y Bioingeniería (BIBE'03), 2003, XP-010637117, ISBN 978-0-7695-1907-4; la publicación internacional n.º WO 01/33436 de Informix Software, Inc.; y la patente estadounidense n.º 5.903.888 a nombre de Cohen et al.

El documento XP-002482800 divulga que los almacenes de datos contienen normalmente tablas grandes y necesitan técnicas tanto para gestionar estas grandes tablas como para ofrecer un buen rendimiento en la realización de consultas a través de estas grandes tablas. La división por intervalos establece correspondencias entre datos y particiones en función de intervalos de valores de clave de partición que se establecen para cada partición. Es el tipo de división más común y se usa frecuentemente con fechas.

Resumen de la invención

La invención está especificada por las reivindicaciones independientes. Las reivindicaciones dependientes incluyen desarrollos adicionales.

Se proporciona un procedimiento a modo de ejemplo para reducir rápidamente el alcance de una búsqueda de información. Mientras que un índice especifica dónde está un elemento de información particular, una correspondencia casi ordenada especifica un intervalo o conjunto de intervalos en los que puede estar un elemento de información particular. A primera vista, parece más útil saber con precisión dónde está ubicado algo en lugar de obtener un conjunto de intervalos en los que puede estar; sin embargo, esa utilidad depende de los costes relativos de usar un índice frente a usar una correspondencia casi ordenada, así como de la distribución de la información y del procedimiento de búsqueda. En determinados casos comunes e importantes, las correspondencias casi ordenadas pueden reducir rápidamente el alcance de una búsqueda en dos o tres órdenes de magnitud, produciendo resultados casi tan rápidos o incluso más rápidos que cuando se usa un índice y con solo una parte del coste computacional y de almacenamiento a la hora de mantener un índice.

Un ejemplo funciona dividiendo un gran espacio de información en muchas extensiones de información más pequeñas. Estas extensiones se anotan con estadísticas acerca de la información que contienen, en particular valores mínimos y máximos (también incluyen otros valores, por ejemplo un cómputo). Cuando una búsqueda de información incluye una restricción basada en valores, los intervalos de valores deseados pueden compararse con el valor mínimo y el valor máximo de cada extensión. Si el intervalo de valores deseado está fuera del intervalo de la extensión, entonces la extensión no puede albergar el valor deseado y, por tanto, no tiene que incluirse en la búsqueda. Si el intervalo deseado se solapa con el intervalo de una extensión particular, entonces esa extensión debe explorarse, incluso aunque no contenga los valores deseados.

En el texto siguiente se usa el término "relación" con sentido inclusivo para incluir tablas, vistas, índices y otras estructuras de datos. Para simplificar la descripción se usa el término "tabla" de manera intercambiable con "relación". También debe observarse que para facilitar la descripción se hace referencia a tipos específicos de almacenamiento tales como "almacenamiento persistente" y "memoria"; las correspondencias casi ordenadas también funcionan correctamente con otros tipos de almacenamiento y combinaciones de tipos.

Más en particular, considérese que en muchas aplicaciones de almacenamiento de datos, un pequeño número de relaciones ocupan el 80% o más del tamaño de la base de datos. Pueden ser, por ejemplo, relaciones que registran en el tiempo eventos reales, eventos que son normalmente muy básicos e importantes para una empresa. En la industria de las telecomunicaciones, estos eventos pueden identificar llamadas telefónicas individuales o componentes de las llamadas telefónicas ("registros de detalles de llamadas"). En lo que respecta a sitios de Internet, estos eventos pueden ser datos que describen clics individuales que han realizado los usuarios. En lo que respecta a empresas que venden productos o servicios a los consumidores, estos eventos pueden ser compras individuales en un punto de venta, o incluso una ruta física que elementos individuales de un carrito de compra pueden haber tomado por toda la tienda. En relación con bancos y corretajes, estos eventos pueden ser transacciones financieras.

En las diferentes industrias y aplicaciones hay ciertas características comunes relacionadas con la manera en que estos eventos se capturan y se usan en un almacén de datos. Cada día se crean, por lo general, cientos de millones de eventos. Normalmente llegan a un almacén de datos agrupados cronológicamente, casi en el orden en que se producen y poco después de haberse producido. Los análisis se llevan a cabo normalmente en secciones cronológicas comparando, por ejemplo, el tráfico en diferentes horas del día o en diferentes ubicaciones geográficas al mismo tiempo. Aunque resulta habitual analizar datos en secciones cronológicas, no es el único ámbito en el que hay una mala ordenación y agrupación. Por ejemplo, las imágenes de los satélites tienden a agruparse y analizarse en bandas longitudinales.

Puesto que la información de estas grandes relaciones es crucial para los intereses de las empresas, la necesidad de almacenar más información es cada vez mayor. Al aumentar la cantidad de información almacenada en una tabla, esto hace que aumente con frecuencia el valor de los análisis, ya que los patrones predictivos tienen más datos que cotejar. Además, la tasa con la que se captura nueva información aumenta un 50% cada año en muchas industrias. Esto significa que estas tablas son cada vez más grandes, variando su tamaño en el tiempo.

Si una empresa registra quinientos millones de eventos al día y desea guardar datos introducidos en línea durante cien días en su almacén de datos, esto requiere guardar cincuenta billones de eventos mantenidos en línea. La naturaleza de la información registrada de estos eventos varía en las diferentes industrias y aplicaciones, pero esto puede necesitar de manera razonable cien bytes por registro. Esto significa que el tamaño de una tabla de eventos puede ser fácilmente de cinco terabytes. Si los cinco terabytes se dividieran de manera equitativa en quinientos discos, cada disco albergaría diez gigabytes de datos de tabla de eventos. Si estos discos se exploraran a una velocidad de 33 megabytes por segundo, cada disco tardaría cinco minutos aproximadamente en leer su parte de toda la tabla de eventos. Esto significa que un almacén de datos con quinientos discos podría gestionar como mucho doce consultas a la hora realizadas en la tabla más importante y central. Evidentemente, esta solución no es eficaz.

El uso de un índice tradicional, tal como un árbol b, mitiga este problema, especialmente cuando solo se van a recuperar algunos registros. Por ejemplo, en un sistema transaccional, si un usuario está recuperando los registros para una gestión específica, entonces un índice de tipo árbol b es muy eficiente. Sin embargo, en lo que respecta a consultas que recuperan entre un número medio y grande de registros, el coste computacional de usar estos índices afecta considerablemente al coste de la consulta. Por ejemplo, algunos distribuidores de sistemas de gestión de bases de datos relacionales (RDBMS) sugieren que no se usen índices si va a accederse a más del 8% de una tabla dada.

Por lo tanto, un ejemplo saca partido de (a) la agrupación cronológica intrínseca de eventos en el tiempo dentro de una tabla de eventos y (b) el deseo común de consultar la tabla de eventos por secciones cronológicas para reducir en varios órdenes de magnitud el tiempo y el coste de recursos de sistema requeridos para explorar la tabla y recuperar los registros solicitados. Cuando esto puede aplicarse, se mejora drásticamente el tiempo de respuesta y el rendimiento. Además, como se ha descrito anteriormente, este ejemplo también puede aplicarse a otras tablas en las que los registros están agrupados intrínsecamente por campos que se usan habitualmente para restringir las cláusulas y/o para ordenar las cláusulas de una consulta.

Además, otro ejemplo puede aplicarse igualmente en relaciones derivadas, especialmente en aquéllas en las que algunos de los valores aparecen físicamente de manera ordenada o casi ordenada. Un ejemplo de esto es una vista materializada ordenada. Otro ejemplo es la relación intermedia creada durante la ordenación de una tabla. En este caso, si se usa una ordenación por sectores para ordenar una relación, entonces los datos de los sectores creados tras la segunda fase de ordenación (la fase de "creación de sectores") están prácticamente ordenados y se benefician del ejemplo mostrado anteriormente. El uso de correspondencias casi ordenadas permite usar la tabla intermedia en muchas operaciones antes de la tercera fase de ordenación ("ordenación final") reduciéndose así considerablemente el tiempo para ordenar de manera eficaz la relación dada. Este uso de una relación parcialmente ordenada puede no estar disponible con los índices tradicionales.

La ventaja principal de un ejemplo en el nivel más abstracto se refiere a reducir la cantidad de tiempo requerida para localizar una cantidad de información relativamente pequeña en un espacio de información muy grande en determinadas situaciones importantes y habituales. Más específicamente, una ventaja se refiere a reducir la parte de un disco de ordenador que debe explorarse para localizar registros de interés.

La técnica más común para conseguir este tipo de ventaja es el uso de índices. Un índice es una estructura de datos individual que correlaciona un conjunto de valores de datos con un conjunto de registros relacionados con esos valores de datos, donde con frecuencia esos valores de datos están incluidos como campos dentro de los registros. Las consultas a las bases de datos utilizan los índices en un proceso de dos etapas. En primer lugar, el índice se consulta para encontrar los registros de datos pertinentes. En segundo lugar se recuperan los registros de datos pertinentes.

El uso simplista de índices de la técnica anterior tiene varios problemas a la hora de mejorar el rendimiento de las consultas, los cuales se evitan mediante los ejemplos mostrados anteriormente. En primer lugar, muchos tipos de índices requieren una cantidad de espacio considerable. En segundo lugar, aunque el tiempo requerido para usar el índice es normalmente una función logarítmica (o una función mejor) del número de registros que está indexándose, esto puede seguir necesitando una cantidad de tiempo considerable para una cantidad de registros muy grande. En tercer lugar, los cambios en el conjunto subyacente de registros requieren cambios correspondientes en el índice, lo que puede ralentizar el tiempo requerido para cargar o filtrar datos. Con cientos de millones de registros entrando y saliendo del sistema cada día, la sobrecarga del mantenimiento de índices adicionales puede ser un grave problema. En cuarto lugar, hay serios problemas con la integridad transaccional y el control de concurrencia que surgen con el uso de índices. En quinto lugar (y probablemente con mayor importancia para un almacén de datos distribuido), cuando el conjunto de registros se divide en muchos discos, no siempre es posible ubicar conjuntamente la parte del índice que correlaciona un valor de datos dado con los registros relacionados con ese valor de datos. Esto significa que se necesita generalmente una comunicación en red entre las etapas de usar el índice para encontrar los registros de datos pertinentes y de explorar después los discos para buscar esos registros de datos. Estas comunicaciones en red adicionales pueden duplicar la latencia de una consulta. Finalmente, los índices requieren configuración, mantenimiento y clasificación para saber qué definiciones de índice serían las más adecuadas para consultas de aplicación con el mínimo coste.

Mientras que el objetivo de un índice es determinar rápidamente la ubicación de un conjunto de registros, el objetivo de una correspondencia casi ordenada es lo opuesto: determinar rápidamente las ubicaciones que no contienen un conjunto de registros. Debido en parte a este enfoque diferente, el ejemplo mostrado anteriormente usa correspondencias casi ordenadas para evitar problemas con los índices. Se necesita muy poco espacio adicional, del orden de 24 bytes por 3 megabytes de datos de registro. Puesto que un ejemplo ubica conjuntamente correspondencias casi ordenadas y los datos de registro que describen, se introduce muy poca latencia computacional adicional. En un sistema con inteligencia en cada uno de los nodos de datos, puesto que (a) las correspondencias casi ordenadas pueden mantenerse y accederse en esa memoria de nodo local y (b) la lógica para acceder a los datos puede estar incluida en ese nodo, el ordenador principal puede ignorar totalmente las correlacionales locales casi ordenadas, dando lugar a un sistema más fiable, más eficiente y de mayor rendimiento.

Las correspondencias casi ordenadas se actualizan de tal manera que mantienen la información de manera precisa, aunque pesimista, y evitan problemas relacionados con el control de la concurrencia y el tiempo de carga adicional. Finalmente, las correspondencias casi ordenadas pueden mantenerse de manera automática, evitando la necesidad de una administración compleja de la base de datos.

Al igual que con las técnicas de almacenamiento de datos tradicionales, un gran espacio de información se divide en una serie de muchos segmentos de información más pequeños denominados extensiones. La idea principal es que cada extensión se anota con el valor máximo y el valor mínimo para tipos particulares de información incluidos en la extensión. Cuando una consulta desea explorar el gran espacio de información para encontrar valores en intervalos particulares, el ejemplo antes mostrado examina primero las anotaciones asociadas a cada extensión para determinar si esa extensión puede contener valores en el intervalo deseado. Las extensiones que no pueden albergar valores en el intervalo deseado no se exploran. Solo se exploran las extensiones que pueden albergar valores en el intervalo deseado. Cuando hay muchos valores de datos agrupados en las extensiones y cuando las consultas son relativamente selectivas en lo que respecta al intervalo de valores que aceptarán, el ejemplo antes mostrado puede eliminar rápidamente gran parte del tiempo requerido para explorar el gran espacio de información.

En un sistema con inteligencia en uno o más nodos de datos, cada nodo inteligente puede mantener las anotaciones para su almacenamiento local, de modo que sus correspondencias de almacenamiento y sus anotaciones pueden mantenerse de manera privada: pueden encapsularse, ser independientes, autónomos y ocultarse de cualquier otro elemento de sistema. El resultado es una obtención muy rápida sin necesidad de mantener un índice centralizado o un índice preciso.

En casos en los que se obtienen algunos registros para ciertas consultas, las correspondencias casi ordenadas pueden aumentar con una correspondencia "raíz" que identifica el nodo o nodos que pueden contener esos registros, de manera que la consulta se envía solamente a los nodos que pueden contener los registros objetivo.

Breve descripción de los dibujos

Estos y otros objetos, características y ventajas de la forma de realización preferida de la invención resultarán evidentes a partir de la siguiente descripción más particular de formas de realización preferidas de la invención, como se ilustra en los dibujos adjuntos, en los que los mismos caracteres de referencia se refieren a las mismas partes a lo largo de las diferentes vistas. Los dibujos no están necesariamente a escala, sino que lo que se pretende es ilustrar los principios de la forma de realización preferida de la invención.

La Fig. 1 es un diagrama de bloques de los componentes de hardware relevantes para la forma de realización preferida de la invención.

La Fig. 2 es un diagrama de bloques de los componentes de software relevantes para la forma de realización preferida de la invención.

La Fig. 3 es un diagrama de bloques de la organización del almacenamiento físico.

La Fig. 4 es un diagrama de bloques de la estructura de datos usada para la tabla y entradas de correspondencias casi ordenadas.

La Fig. 5 es un diagrama de bloques de la estructura de datos usada para indexar entradas de correspondencias casi ordenadas.

La Fig. 6 es un diagrama de flujo de un procedimiento para calcular una lista de extensiones a partir de una expresión sencilla.

La Fig. 7 es un diagrama de flujo de un procedimiento para calcular una lista de extensiones a partir de una combinación lógica de expresiones sencillas.

La Fig. 8 es un diagrama de flujo de un procedimiento para actualizar entradas de correspondencias casi ordenadas.

La Fig. 9 es un diagrama de flujo de un procedimiento para registrar el valor mínimos y el valor máximo para un índice de columna dado de un número de bloques dado de un identificador de tabla dado.

La Fig. 10 es un diagrama de bloques de la estructura de árbol usada para representar expresiones de búsqueda.

Descripción detallada de una forma de realización preferida

La Fig. 1 muestra los componentes de hardware relevantes de la forma de realización preferida de la invención. Se usa una unidad de almacenamiento persistente 100-1 para guardar información. Un procesador de información 100-2 almacena y recupera información de una unidad de almacenamiento persistente 100-1 en la dirección de un procesador principal 100-3. Un procesador principal 100-3 recibe solicitudes desde, y contesta a, un procesador cliente 100-4. Un procesador de información 100-2 comprende además una memoria 100-2-1, una interfaz de red 100-2-2 y un procesador de propósito general 100-2-3. Un procesador principal 100-3 consiste además en una memoria 100-3-1, una interfaz de red 100-3-2 y un procesador de propósito general 100-3-3.

En un modo de funcionamiento, el espacio de una unidad de almacenamiento persistente 100-1 está dividido en varias particiones, incluyendo una partición primaria para almacenar información de usuario, una partición espejo para almacenar una copia de seguridad de la información primaria de otras unidades de almacenamiento persistentes 100-1, una partición temporal para guardar resultados intermedios en la dirección de un procesador de información 100-2, y una partición central para guardar información acerca de la información de diseño de la unidad de almacenamiento persistente 100-1. En un modo de funcionamiento, el tamaño de la partición primaria es de 15 gigabytes aproximadamente, pero también pueden usarse otros tamaños.

La Fig. 2 muestra los componentes de software relevantes de la forma de realización preferida de la invención. Un gestor de almacenamiento 200-1 se ejecuta en un procesador de información 100-2. Un analizador de solicitudes de cliente 200-10, un generador de códigos de ejecución de solicitud 200-11, un gestor de catálogo 200-12, un generador de estadísticas 200-13 y un analizador de calificación 200-14 se ejecutan en un procesador principal 100-3.

La Fig. 3 ilustra la división de un gran espacio de información en segmentos más pequeños. Todo el espacio 300-1 está dividido en un gran número de extensiones más pequeñas de tamaño fijo 300-2. Cada extensión está subdividida además en una pluralidad de bloques de subextensión contiguos de tamaño fijo 300-3. Es posible que una extensión 300-2 o un bloque 300-3 tengan un tamaño variable en lugar de un tamaño fijo, lo que no afecta apenas a la forma de realización preferida de la invención. En una forma de realización alternativa de la invención, los tamaños de bloque pueden ser variables y pueden modificarse dinámicamente en función de, por ejemplo, el tipo de atributo con el que están relacionadas las anotaciones. Por ejemplo, si los valores mínimos y máximos se refieren

a datos de indicación de tiempo, el tamaño de la extensión puede modificarse de manera que todos los datos de un periodo de tiempo particular se almacenen en un bloque de datos.

La ventaja principal de usar tamaños fijos es que determinados cálculos se vuelven más rápidos y menos complejos, por ejemplo la correlación de números de bloque con direcciones de disco. En una forma de realización alternativa de la invención, el tamaño fijo también puede cambiar dinámicamente en función del tamaño y de otras características de la base de datos global.

Una extensión 300-2 es una unidad práctica de asignación de almacenamiento persistente, de manera que a medida que crece un espacio de información 300-1, su crecimiento se cuantifica en unidades de extensiones 300-2. Aunque el espacio de direcciones dentro de una extensión 300-2 es contiguo, sin orificios, un espacio de información 300-1 consiste en extensiones 300-2 posiblemente no contiguas. En particular, no es necesario que una asignación posterior de una extensión 300-2 deba generar un intervalo de direcciones que permita cualquier relación fija con el intervalo de direcciones de cualquier asignación anterior.

Un bloque 300-3 es la unidad más pequeña de transferencia entre la unidad de almacenamiento persistente 100-1 que alberga todo el espacio 300-1 y un procesador de información 100-2 que puede examinar y manipular la información. En una forma de realización, un bloque 300-3 ocupa 128 kilobytes, pero también pueden usarse otros tamaños.

En una forma de realización, una extensión 300-2 consiste en 24 bloques 300-3, de manera que una extensión 300-2 tiene un tamaño de 3 megabytes, pero también pueden utilizarse otros tamaños y números de bloques 300-3. Los bloques de una extensión están numerados secuencialmente en un orden de dirección creciente. Nueva información se introduce en una extensión 300-2 en un orden de número de bloque creciente, específicamente al final del bloque 300-3 con el número más alto que está en uso.

La Fig. 4 muestra la estructura de una tabla de correspondencia casi ordenada 400, que consiste en entradas consecutivas de tabla de correspondencia casi ordenada 410. Cada entrada de tabla de correspondencia casi ordenada 410 consiste en un identificador de tabla 410-1, un índice de columna 410-2, un valor de datos mínimo 410-3, un valor de datos máximo 410-4 y un identificador de extensión 410-5.

El identificador de tabla 410-1 designa de manera única un espacio de información 300-1. En una forma de realización es un valor numérico de 4 bytes que identifica de manera única una tabla relacional.

El índice de columna 410-2 identifica de manera única una clase particular de información dentro del espacio de información 300-1 identificado por el identificador de tabla 410-1. En una forma de realización, un índice de columna 410-2 es un valor numérico de 2 bytes que denota una columna de la tabla relacional identificada por el identificador de tabla 410-1. En una forma de realización, el valor del índice de columna corresponde al orden en que las columnas están definidas en la tabla relacional, de tal manera que el valor 0 denota la primera columna definida en la tabla.

En una forma de realización de la invención, el valor de datos mínimo 410-1-3 y el valor de datos máximo 410-1-4 son cantidades de 8 bytes que pueden albergar tipos diferentes de valores de datos, incluyendo fechas, horas, fecha y hora o valores enteros. Los tipos reales de datos determinados por el valor de datos mínimo 410-1-3 y el valor de datos máximo 410-1-4 están especificados en la definición de la columna que está denotada con el índice de columna 410-2.

En una forma de realización de la invención, el identificador de extensión 410-5 es un valor numérico de 2 bytes que designa una extensión particular del espacio de información 300-1 designado por el identificador de tabla 410-1. Particiones de almacenamiento más grandes usarán o bien identificadores de extensión de 4 bytes o extensiones más grandes 300-2. Por tanto, en esta forma de realización de la invención, cada entrada de tabla de correspondencia casi ordenada 410 comprende 24 bytes.

En una forma de realización de la invención, las entradas de tabla de correspondencia casi ordenadas 410 están agrupadas por el índice de columna 410-2, de manera que todas las entradas de la enésima columna de una tabla están agrupadas conjuntamente en un único bloque 300-3. Debe observarse que cuando la cantidad total de almacenamiento de información disponible es de 15 gigabytes aproximadamente, el número máximo de extensiones para este almacenamiento es de 5154 aproximadamente. Cuando el tamaño de una entrada de tabla de correspondencia casi ordenada 410 es de 24 bytes, es posible agrupar las entradas de tabla de correspondencia casi ordenada 410 de cada una de las posibles 5154 extensiones en un único bloque 300-3 de 128 K. Puesto que un bloque es la unidad de transferencia entre la unidad de almacenamiento persistente 100-1 y el procesador de información 100-2, es posible acceder a toda la correspondencia casi ordenada de una columna dada en una única transferencia desde la unidad de almacenamiento persistente 100-1.

Como optimización, es posible agrupar las entradas de tabla de correspondencia casi ordenada 410 del mismo índice de columna de todas las tablas en un único bloque 300-3.

La Fig. 5 muestra la estructura de un índice de correspondencia casi ordenada 500, que consiste en entradas consecutivas de índice de correspondencia casi ordenada 510. Cada entrada de índice de correspondencia casi ordenada 510 consiste en un identificador de tabla 510-1, un índice de columna 510-2 y un número de bloque 510-3. El identificador de tabla 510-1 debe tener la misma función e interpretación que el identificador de tabla 410-1 de las entradas de tabla de correspondencia casi ordenada. En una forma de realización es un valor numérico de 4 bytes que identifica de manera única una tabla relacional. El índice de columna 510-2 debe tener la misma función e interpretación que el índice de columna 410-2 de las entradas de tabla de correspondencia casi ordenada. En una forma de realización, es un valor numérico de 2 bytes. El número de bloque 510-3 designa el bloque 300-3 que contiene las entradas de tabla de correspondencia casi ordenada 410 para la columna de tabla designada por el identificador de tabla 510-1 y el índice de columna 510-2. En una forma de realización debe ser al menos un valor numérico de 2 bytes.

Cuando un gran espacio de información 300-1 está dividido en un conjunto de extensiones 300-2, una tabla de correspondencia casi ordenada 400 puede consultarse para determinar el subconjunto de extensiones 300-2 que puede contener posiblemente valores de interés. Las extensiones 300-2 que no contienen posiblemente valores de interés no tienen que explorarse. De manera conceptual, el algoritmo principal de la forma de realización preferida de la invención utiliza una descripción de información que recuperar y devuelve una lista de extensiones que podrían contener esa información. En la práctica, esta tarea se divide en varios algoritmos que se ejecutan en diferentes componentes de hardware y software.

La Fig. 6 muestra un diagrama de flujo de un procedimiento para calcular la lista de identificadores de extensión para expresiones sencillas de correspondencias casi ordenadas. Una expresión sencilla de correspondencias casi ordenadas es una que compara un valor de una columna particular con una constante literal. Seis comparaciones permitidas son: mayor que, menor que, mayor o igual que, menor o igual que, igual a y diferente de. Las cuatro entradas del procedimiento son (1) un identificador de tabla que se usa para denotar el espacio de información objetivo 300-1, (2) un índice de columna que se usa para denotar una clase de espacio de información especificada por el identificador de tabla, (3) un valor objetivo de un tipo compatible con la clase de información asociada al índice de columna, y (4) un operador de comparación, tal como mayor que o menor que, con el que comparar el valor objetivo y el valor mínimo o máximo de la clase denotada de información en las extensiones asociadas. El procedimiento se ejecuta en el componente de gestión de almacenamiento 200-1 del procesador de información 100-2.

En la etapa 600 se inicializan variables locales, incluyendo una lista de resultados de identificadores de extensión. En la etapa 610, el gestor de almacenamiento 200-1 encuentra la entrada de índice de correspondencia casi ordenada 510 cuyo índice de columna 510-2 coincide con el índice de columna que se pasa al procedimiento. En la etapa 615, si la entrada de índice 510 encontrada en la etapa 610 no es útil (su bandera de utilidad 510-4 es falsa), entonces el proceso devuelve una lista de todos los identificadores de extensión en la etapa 618. En otro caso, en la etapa 620, el gestor de almacenamiento 200-1 encuentra el bloque 300-3 de la tabla de correspondencia casi ordenada 400 indicado por el número de bloque 510-3 en la entrada de índice de correspondencia casi ordenada 510 encontrada en la etapa 610. En una forma de realización, este bloque 300-3 se almacenará en la memoria 100-2-1 después de leerse por primera vez desde una unidad de almacenamiento persistente 100-1.

La etapa 630 entra en un bucle por cada entrada de tabla de correspondencia casi ordenada 410 del bloque 300-3 obtenido en la etapa 620. La condición de la etapa 640 comprueba si (a) el identificador de tabla 410-1 de la siguiente entrada de tabla de correspondencia casi ordenada 410 es igual al identificador de tabla introducido en el procedimiento, y si (b) el índice de columna 410-2 de la siguiente entrada de tabla de correspondencia casi ordenada 410 es igual al índice de columna introducido en el procedimiento. Si se cumplen ambas condiciones, entonces la etapa 650 comprueba si el operador de comparación introducido en el procedimiento es un operador 'mayor que' y, de ser así, la etapa 660 comprueba si el valor objetivo introducido en el procedimiento es mayor que el valor de datos mínimo 410-3 asociado a la siguiente entrada de tabla de correspondencia casi ordenada 410. Si es así, el identificador de extensión 410-5 de la siguiente entrada de tabla de correspondencia casi ordenada 410 se añade a una lista de resultados de identificadores de extensión en la etapa 680. Si el operador de comparación es, en cambio, un operador 'menor que', entonces la etapa 670 comprueba si el valor objetivo introducido en el procedimiento es menor que el valor de datos máximo 410-4 asociado a la siguiente entrada de tabla de correspondencia casi ordenada 410. Si es así, el identificador de extensión 410-5 de la siguiente entrada de tabla de correspondencia casi ordenada 410 se añade a una lista de resultados de extensiones en la etapa 680.

Un procesamiento similar se produce para los operadores de comparación '>=', '<='. Para procesar el operador de comparación ‘igual a’, la extensión 300-2 se incluye si la constate objetivo está dentro del intervalo del valor mínimo y el valor máximo. Para procesar el operador de comparación ‘diferente de’, la extensión 300-2 no se tiene en cuenta si y solo si sus valores mínimo y máximo son iguales a la constante objetivo. Después de que todas las entradas de tabla de correspondencia casi ordenada 410 del bloque 300-3 obtenido en la etapa 620 se hayan examinado, la lista de resultados de identificadores de extensión se devuelve como el resultado del procedimiento en la etapa 690.

El procedimiento descrito anteriormente es útil para situaciones sencillas en las que la información solicitada puede describirse como una comparación sencilla de un único valor de columna con un valor constante. Pero la mayoría de solicitudes de información tienen condiciones más complejas. Algunas de estas pueden describirse como una combinación lógica de las comparaciones simples descritas anteriormente. Por ejemplo, una solicitud para comparar resultados anuales puede recuperar eventos que se produjeron durante el mes de enero de este año o durante el mes de enero del año pasado. Esto puede expresarse como una combinación lógica:

((valor-columna-fecha >= '01/01/03') Y (valor-columna-fecha <'02/01/03')) O
((valor-columna-fecha >= '01/01/02') Y (valor-columna-fecha <'02/01/02'))

La Fig. 7 muestra un diagrama de flujo de un procedimiento para devolver una lista de identificadores de extensión 300-2 que puede contener registros que satisfacen una combinación lógica de comparaciones sencillas. La idea básica del procedimiento es calcular la lista de identificadores de extensión para cada una de las comparaciones de la combinación lógica y después formar la intersección de las listas resultantes para cada conjunción lógica y formar la unión de las listas resultantes para cada disyunción lógica. La lista resultante incluirá identificadores de solo aquellas extensiones que pueden contener valores que satisfagan la combinación lógica en su totalidad.

El procedimiento acepta como entrada un árbol de expresión de correspondencias casi ordenadas 1000 y genera como salida un conjunto de identificadores de extensión. El procedimiento actúa de manera recursiva primero en profundidad. Cuando encuentra un nodo hoja que contiene una expresión sencilla de correspondencias casi ordenadas 1000-3, transfiere los contenidos de la expresión al procedimiento descrito en la Fig. 6 para calcular el conjunto de identificadores de extensión para expresiones sencillas de correspondencias casi ordenadas. Algunos nodos hoja 1000-2 contienen una indicación de que son equivalentes a todos los identificadores de extensión. Esto puede ser el caso si la solicitud de recuperación usó una expresión que era muy compleja de manejar por una correspondencia casi ordenada, por ejemplo cuando se compara un valor de columna con otro valor de columna. Cuando el procedimiento encuentra un nodo de este tipo, devuelve el conjunto de todos los identificadores de extensión. Cuando el procedimiento encuentra un nodo que no es hoja 1000-1 se llama recursivamente a sí mismo en cada uno de los hijos del nodo que no es hoja con el fin de calcular sus conjuntos de identificadores de extensión. Después combina estos conjuntos de identificadores de extensión según el operador lógico especificado en el nodo que no es hoja. Si el operador es una conjunción (Y), entonces el procedimiento forma un nuevo conjunto de identificadores de extensión que es la intersección de los conjuntos de identificadores de extensión devueltos para cada uno de los hijos del nodo. Si el operador es una disyunción (O), entonces el procedimiento forma un nuevo conjunto de identificadores de extensión que es la unión de los conjuntos de identificadores de extensión devueltos para cada uno de los hijos del nodo.

El árbol de expresión de correspondencias casi ordenadas usado como entrada al procedimiento descrito anteriormente se genera en el procesador principal 100-3 mediante el analizador de calificación 200-14. Cuando el analizador de solicitud de cliente 200-10 recibe una solicitud para recuperar información, analiza la especificación de la información a recuperar. En una forma de realización, el analizador de consulta de progreso se usa para generar un árbol de expresión que corresponde a la cláusula 'donde' de una consulta SQL. Algunas de las restricciones especificadas en la solicitud de información pueden ser muy complejas de manejar por una correspondencia casi ordenada. El analizador de calificación 200-14 transforma el árbol de expresión producido por el analizador de solicitud de cliente 200-10 en la forma de árbol de expresión de correspondencias casi ordenadas 1000 usada anteriormente. Estos datos se envían después junto con la solicitud de información desde un procesador principal 100-3 a uno o más procesadores de información 100-2, que usan el árbol de expresión de correspondencias casi ordenadas 1000 descrito en el procedimiento anterior para limitar el alcance de una búsqueda de información. En una forma de realización, el analizador de calificación 200-10 lleva a cabo determinadas optimizaciones en este proceso de transformación. Si un nodo es una disyunción (O) y si alguno de sus hijos es un indicador de tipo 'todas las extensiones', entonces es posible sustituir la disyunción por el indicador ‘todas las extensiones’ y podar los hijos. Si un nodo es una conjunción (Y) y si todos sus hijos tienen el indicador 'todas las extensiones', entonces es posible sustituir la conjunción por el indicador ‘todas las extensiones’ y podar los hijos. Si un nodo es una conjunción (Y) y si algunos de sus hijos tienen el indicador 'todas las extensiones' mientras que otros de sus hijos son expresiones sencillas de correspondencias casi ordenadas, entonces es posible podar los hijos que tengan el indicador 'todas las extensiones' dejando intactos los otros hijos.

A medida que cambia la información almacenada en una extensión 300-2, también puede ser necesario cambiar las anotaciones de valor mínimo 410-3 y de valor máximo 410-4 de la entrada de tabla de correspondencia casi ordenada 410 asociada a esa extensión 300-2. Si fuera necesario, en un modo de funcionamiento estos cambios pueden producirse al mismo tiempo que cambia la información subyacente, de manera gradual. En otro modo de funcionamiento preferido, los cambios producidos en la tabla de correspondencia casi ordenada pueden realizarse por lotes después de modificar la información subyacente.

El enfoque del modo por lotes relacionado con las actualizaciones de las correspondencias casi ordenadas se lleva a cabo en dos fases. En primer lugar, se generan estadísticas para cada extensión 300-2 de un espacio de información 300-1. Después se determina si el tamaño y la distribución de información dentro del espacio de

información 300-1 es tal que las solicitudes relacionadas con esta información se beneficiarán de usar una correspondencia casi ordenada.

La Fig. 8 muestra un diagrama de flujo de un procedimiento usado para actualizar correspondencias casi ordenadas. Las entradas del procedimiento incluyen (1) un identificador de tabla que designa un espacio de información particular 300-1, (2) una descripción del tamaño y diseño de un registro del espacio de información 300-1, y (3) una lista de índices de columna que pueden ser candidatos a correspondencias casi ordenadas. En una forma de realización, cualquier columna cuyo tipo de datos sea una fecha, una hora, una fecha y hora o un valor entero es un candidato a correspondencia casi ordenada; sin embargo, también es razonable permitir correspondencias casi ordenadas para otros tipos de datos. El efecto del procedimiento es crear o actualizar entradas de tabla de correspondencia casi ordenada 410 para las extensiones 300-2 del espacio de información 300-1 designado por la entrada de identificador de tabla. La salida del procedimiento no es esencial para el funcionamiento de la forma de realización preferida de la invención.

El procedimiento comienza en la etapa 800 asignando e inicializando el almacenamiento para guardar un valor mínimo y un valor máximo para cada columna de la lista de entrada de índices de columna. Después, recorre en un bucle cada bloque 300-3 asociado a la entrada de identificador de tabla, leyendo el siguiente bloque 300-3 de la unidad de almacenamiento persistente 100-1 en la etapa 805. El procedimiento entra en un bucle por cada registro almacenado en el bloque 300-3, localizando el siguiente registro en la etapa 810. En registros de tamaño fijo, el procedimiento usado en la etapa 810 para localizar el siguiente registro es simplemente incrementar un puntero de registro con el tamaño fijo de un registro. En registros de tamaño variable, el procedimiento usado en la etapa 810 para localizar el siguiente registro implica encontrar un campo del registro actual que especifique el tamaño del registro actual y después incrementar el puntero de registro en ese tamaño. Después, el procedimiento entra en un bucle por cada índice de columna de la lista de entrada de índices de columna, obteniendo el siguiente índice de columna en la etapa 815. En la etapa 820, el procedimiento extrae el valor actual del campo del siguiente registro que está ubicado en el siguiente índice de columna.

En la etapa 825, el procedimiento comprueba si el valor actual extraído en la etapa 820 es inferior al valor mínimo de la siguiente columna que se asignó e inicializó en la etapa 800. Si es así, la etapa 828 fija el valor mínimo para la siguiente columna igual al valor actual extraído en la etapa 820. En caso contrario, en la etapa 830, el procedimiento comprueba si el valor actual extraído en la etapa 820 es mayor que el valor máximo de la siguiente columna que se asignó e inicializó en la etapa 800. Si es así, la etapa 833 fija el valor máximo de la siguiente columna igual al valor actual extraído en la etapa 820. En la comparación 835, si hay más índices de columna en la lista de entrada de índices de columna, el procedimiento vuelve a la etapa 815. En caso contrario, en la comparación 838, si hay más registros en el siguiente bloque, el procedimiento vuelve a la etapa 810. Si no, el procedimiento ha finalizado la recopilación de los valores mínimos y máximos de todas las columnas de interés de un bloque 300-3. En la etapa 840, el procedimiento entra en bucle para el almacenamiento asignado en la etapa 800 para guardar los valores mínimo y máximo asociados a cada índice de columna de la lista de entrada de índices de columna, recuperando el par de valores mínimo y máximo asociado al siguiente índice de columna.

En la etapa 845, para cada dicho par de valores, el procedimiento llama a una subrutina descrita en la Fig. 9 para registrar el valor mínimos y el valor máximo para un índice de columna dado de un número de bloque dado de un identificador de tabla dado. Si en la etapa 850 hay más índices de columna que procesar, el procedimiento vuelve a la etapa 840. En caso contrario, si hay más bloques que leer en la etapa 855, el procedimiento vuelve a la etapa 805. Si no, el procedimiento finaliza y sale.

En una forma de realización de la invención, el código para implementar el procedimiento descrito anteriormente en la Fig. 8 se genera en el procesador principal 100-3 mediante un generador de código de ejecución de solicitud 200-11. Cuando un administrador o aplicación de una base de datos solicita la recopilación de nuevas estadísticas de correspondencias casi ordenadas para una tabla identificada, un generador de estadísticas 200-13 consulta un gestor de catálogo 200-12 para determinar qué índices de columna de la tabla identificada son candidatos a correspondencia casi ordenada. Esta determinación está basada en el tipo de datos de la columna. En una forma de realización, las columnas cuyo tipo de datos en una fecha, una hora, una fecha y hora o un entero, son candidatas a correspondencia casi ordenada. Después, el generador de estadísticas 200-13 invoca al generador de código de ejecución de solicitud 200-11 para generar el código que implementa el procedimiento descrito anteriormente en la Fig. 7, pero con ciertas modificaciones en el procedimiento descrito en la Fig. 7.

Puesto que el número de columnas que son candidatas a la generación de correspondencias casi ordenadas se conoce cuando se genera el código, la asignación y la inicialización del almacenamiento en la etapa 700 puede sustituirse por la declaración y la inicialización de variables apiladas que albergan los valores mínimos y máximos de cada columna de interés de cada bloque 300-3. Además, puesto que el número de columnas candidatas se conoce cuando se genera el código, los bucles que recorren las columnas empezando por las etapas 715 y 740 pueden simplificarse, y consisten en código que procesa el conjunto conocido y fijo de índices de columnas.

La Fig. 9 es un diagrama de flujo de un procedimiento para generar nuevas entradas de correspondencias casi ordenadas. El procedimiento toma como entrada (1) un identificador de tabla, (2) un índice de columna, (3) un

número de bloque, (4) un valor mínimo y (5) un valor máximo. El identificador de tabla designa un espacio de información 100-1 y, en una forma de realización, una tabla de base de datos relacional. El índice de columna designa una clase de información en el espacio de información 100-1 y, en una forma de realización, una columna de una tabla de base de datos relacional. El número de bloque designa un bloque 300-3 del espacio de información 100-1 y, en una forma de realización, un conjunto particular de 128 Kilobytes que contienen registros de una tabla de base de datos relacional.

Los valores mínimos y máximos son valores de datos del tipo de datos asociado a la clase de información designada por el índice de columna y son cantidades de 8 bytes en una forma de realización. El efecto del procedimiento es crear o actualizar potencialmente entradas de tabla de correspondencia casi ordenada 410 y entradas de índice de correspondencia casi ordenada 510. Las salidas del procedimiento, si las hubiera, no son relevantes para la forma de realización preferida de la invención.

El procedimiento lleva a cabo algún procesamiento especial para el primer y el último número de bloque de una extensión 300-2, y para el último número de bloque de una tabla. En lo que respecta al primer número de bloque de una extensión 300-2, el procedimiento inicializa un valor mínimo y un valor máximo de ejecución para cada columna candidata de la extensión 300-2. En lo que respecta a números de bloque entre el primer y el último número de bloque de una extensión 300-2, el procedimiento actualiza el valor mínimo y el valor máximo de ejecución para incluir el valor mínimo y el valor máximo de entrada. En lo que respecta al último número de bloque de una extensión 300-2, el procedimiento crea un registro temporal de los valores mínimos y máximos de todas las columnas candidatas de la extensión 300-2.

Tras procesar el último número de bloque de una tabla, el procedimiento aplica una política para determinar si la cantidad de datos y la distribución de valores almacenados en los registros temporales justifican una correspondencia casi ordenada para las columnas candidatas de la tabla. En lo que respecta a cada columna candidata que debería tener una correspondencia casi ordenada con dicha política, el procedimiento actualiza entradas de tabla de correspondencia casi ordenada para cada extensión asociada a la columna candidata para incluir el valor mínimo y el valor máximo para esa columna y extensión, como se ha calculado mediante el procedimiento anterior.

En una forma de realización de la invención, la política es proporcionar siempre una correspondencia casi ordenada a cada columna candidata. De esta manera, una tabla cuya correspondencia casi ordenada mejoró en el tiempo, a través de una mejor ordenación de nuevos registros, puede gestionarse bien. En una forma de realización relacionada, la entrada de índice de correspondencia casi ordenada 510 puede ampliarse para incluir una bandera de utilidad. Si la distribución de valores para una columna dada de una tabla dada se distribuyó ampliamente a través de las extensiones 300-2 para proporcionar una correspondencia casi ordenada útil, entonces la entrada de índice 510 para el identificador de tabla dado 510-1 y el índice de columna 510-2 pueden tener su bandera de utilidad fijada a 'falso'. Si la distribución de valores mejoró en el tiempo, una política puede dictaminar el momento en que la bandera de utilidad se fijó a 'verdadero', de modo que la correspondencia casi ordenada se usará para solicitudes subsiguientes de información en función de esa columna.

En otra forma de realización de la invención se usa una política para determinar si una tabla es lo bastante grande como para justificar correspondencias casi ordenadas para sus columnas. Las tablas que son lo bastante pequeñas como para almacenarse en una única extensión no se beneficiarán de una correspondencia casi ordenada siempre que se devuelvan algunos datos de una solicitud de información. El tiempo requerido para usar una correspondencia casi ordenada para tales casos se desperdiciaría, lo que aumentaría el tiempo total requerido para satisfacer la solicitud. Por el contrario, las consultas a tablas muy grandes son las que más se benefician del uso de correspondencias casi ordenadas.

Si el objetivo es mejorar el tiempo de respuesta de todas las consultas, entonces puede tener sentido usar correspondencias casi ordenadas para tablas más grandes que el tamaño de una extensión, siempre que la distribución de los valores de columna ofrezcan una buena especificación. Sin embargo, si el objetivo es mejorar el rendimiento total de todas las consultas, entonces es probable que el 90% o más del tiempo ahorrado mediante el uso de correspondencias casi ordenadas se deba al uso de correspondencias casi ordenadas para consultas a tablas muy grandes. En este último caso podría elegirse una política para definir correspondencias casi ordenadas solo para tablas que ocupen más del 2% del almacenamiento disponible, o más de 300 megabytes cuando hay 15 gigabytes de almacenamiento disponible. Estos números y porcentajes no son esenciales para el funcionamiento de la forma de realización preferida de la invención, pudiendo usarse otras políticas.

En otra forma de realización adicional de la invención se usa una política para determinar si la distribución de valores de columna a través de las extensiones 300-2 justifica el uso de una correspondencia casi ordenada. Si cada extensión 300-2 de un espacio de información 300-1 tenía precisamente los mismos valores mínimo y máximo para un índice de columna dado, entonces no sería beneficioso usar una correspondencia casi ordenada siempre que algún dato satisfaga la solicitud. En este caso, el 100% de los intervalos de valores se solapa con el 100% de las extensiones 300-2. Cuanto menor solapamiento haya entre los intervalos de valores de las extensiones 300-2, mayor será el beneficio de usar correspondencias casi ordenadas.

Para valorar la distribución de valores de columna en esta forma de realización, el procedimiento comprueba el intervalo de valores de columna a través de las extensiones 300-2 de un espacio de información 300-1. Si los intervalos de dos extensiones 300-2 se solapan en más del 50%, el procedimiento los junta en un único 'sector' lógico. Si los intervalos de dos extensiones 300-2 se solapan en menos del 50%, el procedimiento los sitúa en diferentes 'sectores' lógicos.

Tras colocar todos los intervalos de extensiones en 'sectores' lógicos, si todas las extensiones 300-2 están en un 'sector' lógico, la política es evitar definir una correspondencia casi ordenada para esa columna. Si el número de 'sectores' lógicos es al menos el 50% del número de extensiones 300-2, la política es definir una correspondencia casi ordenada. Si algún 'sector' lógico contiene más del 50% de los datos del espacio de información 300-1, la política evita definir una correspondencia casi ordenada; en caso contrario, la política es definir una correspondencia casi ordenada. Estos porcentajes no son esenciales para el funcionamiento de la forma de realización preferida de la invención, pudiendo usarse otras políticas.

En el intervalo entre actualizaciones por lotes de correspondencias casi ordenadas, la creación, el borrado y la modificación de información puede afectar a la validez de las correspondencias casi ordenadas. La forma de realización preferida de la invención adopta un enfoque conservador y pesimista en tales situaciones, como se describe en las siguientes secciones.

Cuando se borra un registro de un espacio de información 100-1 denotado mediante un identificador de tabla, si ese registro contenía un valor de columna que era el valor mínimo para todas las columnas de la extensión 300-2 del registro, entonces el nuevo valor mínimo para esa columna de esa extensión 300-2 puede ser mayor que el valor mínimo registrado 410-3 en la entrada de tabla de correspondencia casi ordenada 410 para esa columna, identificador de tabla y extensión 300-2. Asimismo, si ese registro contenía un valor de columna que era el valor máximo para todas las columnas de la extensión 300-2 del registro, entonces el nuevo valor máximo para esa columna de esa extensión 300-2 puede ser menor que el valor máximo registrado 410-4 en la entrada de tabla de correspondencia casi ordenada 410 para esa columna, identificador de tabla y extensión 300-2. Dicho de otro modo, borrar un registro puede tener el efecto de acotar el intervalo de valores para las columnas de una extensión 300-2.

Si la entrada de correspondencia casi ordenada 410 para esa extensión 300-2 se actualizara para reflejar tales nuevos valores mínimos o máximos, esto podría ayudar a mejorar el rendimiento de solicitudes posteriores, lo que podría evitar explorar la extensión 300-2 definida de manera más acotada. Sin embargo, en el modo de funcionamiento por lotes, el borrado de registros no afecta a las entradas de tabla de correspondencia casi ordenada 410. Si el intervalo de la extensión 300-2 se acotó modificando la entrada de tabla de correspondencia casi ordenada 410 correspondiente, entonces podrían surgir problemas de control de concurrencia si el borrado formó parte de una transacción que fue abortada. Además, si el espacio de información 100-1 soporta múltiples versiones, entonces determinadas transacciones o aplicaciones que funcionan con una versión anterior necesitarían tener acceso a los registros borrados y no podrían acceder a los mismos si la entrada de tabla de correspondencia casi ordenada 410 no tuviera asignada una versión.

Un tratamiento más liberal y optimista de los registros borrados es posible. Si la tabla de correspondencia casi ordenada y la información de índice de correspondencia casi ordenada tienen asignadas una versión junto con los espacios de información que describen y si la tabla de correspondencia casi ordenada y la información de índice de correspondencia casi ordenada se actualizan usando el mismo mecanismo de transacción como parte de la misma transacción que borra un registro, entonces puede suponerse de manera razonable que se consigue un buen rendimiento. El coste de esta posible mayor precisión de la correspondencia casi ordenada es un procesamiento añadido en cada borrado de registro. Si la correspondencia casi ordenada ya es específica y útil, entonces el beneficio de hacerla incluso más específica y útil no puede justificar el coste computacional añadido en cada borrado de registro.

Cuando se crea un nuevo registro, si ese nuevo registro contenía un valor de columna que era el valor mínimo para todas las columnas de la extensión 300-2 del registro, entonces el nuevo valor mínimo 410-3 de la entrada de tabla de correspondencia casi ordenada 410 asociada a esa columna para esa extensión 300-2 debe reducirse al menos al valor especificado en el nuevo registro. Asimismo, si ese nuevo registro contenía un valor de columna que era el valor máximo para todas las columnas de la extensión 300-2 del registro, entonces el nuevo valor máximo 410-4 de la entrada de tabla de correspondencia casi ordenada 410 asociada a esa columna para esa extensión 300-2 debe aumentarse a al menos el valor especificado en el nuevo registro. Dicho de otro modo, crear un registro puede tener el efecto de ampliar el intervalo de valores para las columnas de una extensión 300-2.

Si las entradas de correspondencias casi ordenadas 410 para la extensión 300-2 del nuevo registro no se actualizan para reflejar dichos nuevos valores mínimos o máximos, entonces las solicitudes de información podrían omitir erróneamente los nuevos registros. En una forma de realización, las entradas de tabla de correspondencia casi ordenada 410 para extensiones 300-2 que contienen registros recién creados se amplían lo máximo posible para abarcar todos los valores posibles del tipo de datos de la columna. Esto es más conservador y pesimista que el caso

de expandir el intervalo de una entrada de tabla de correspondencia casi ordenada 410 de la extensión 300-2 para abarcar solamente los nuevos valores mínimos y máximos especificados en el nuevo registro.

Si el gestor de almacenamiento 200-1 asigna una nueva extensión 300-2 para albergar un registro recién creado, también crea nuevas entradas de tabla de correspondencia casi ordenada 410 para cada columna sobre la que va a definirse una correspondencia casi ordenada. En una forma de realización, el valor mínimo 410-3 y el valor máximo 410-4 asociados a estas entradas de tabla de correspondencia casi ordenada 410 tienen el intervalo más amplio posible. Si el registro recién creado es el primer registro asociado al espacio de información 100-1 (o a una tabla de base de datos relacional en una forma de realización), entonces el gestor de almacenamiento 200-1 también crea una entrada de índice de correspondencia casi ordenada 510 para cada nueva entrada de tabla de correspondencia casi ordenada 410. Inicializa estas entradas de índice 510 con el número de bloque que contiene las nuevas entradas de tabla de correspondencia casi ordenada 410.

A modo de optimización, cuando las entradas de tabla de correspondencia casi ordenada 410 para un número de columna dado a través de todas las tablas se introducen en el mismo bloque 300-3, entonces una entrada de índice de correspondencia casi ordenada 510 se crea para un índice de columna dado solo si no hay otra entrada de índice de correspondencia casi ordenada 510 ya definida para ese índice, independiente del identificador de tabla.

Un tratamiento más liberal y optimista de registros recién creados es posible. Si la tabla de correspondencia casi ordenada y la información de índice de correspondencia casi ordenada tienen asignadas una versión junto con los espacios de información que describen, y si la tabla de correspondencia casi ordenada y la información de índice de correspondencia casi ordenada se actualizan usando el mismo mecanismo de transacción y como parte de la misma transacción que crea un nuevo registro, entonces puede suponerse de manera razonable que se obtiene un buen rendimiento. El coste de esta posible mayor precisión de la correspondencia casi ordenada es un procesamiento añadido en cada creación de registro. En entornos en los que se crean cientos de millones de nuevos registros en un proceso de carga por lotes, el coste computacional de ajustar gradualmente valores mínimos 410-3 y máximos 410-4 de entradas de tabla de correspondencia casi ordenada 410 puede ser inaceptable. En tales casos de carga por lotes, la generación de estadísticas por lotes, como la descrita anteriormente en la Fig. 7, puede ser más eficiente desde un punto de vista computacional.

En otra forma de realización, los valores reales en registros recién creados se usan para actualizar el valor mínimo 410-3 y el valor máximo 410-4 de la entrada de tabla de correspondencia casi ordenada 410 correspondiente. Esto se realiza sin crear múltiples versiones de los datos de correspondencias casi ordenadas o deshaciendo los cambios de la entrada de tabla de correspondencia casi ordenada 410 en caso de abortar la transacción en la que se crean los nuevos registros. El efecto de esto es que el intervalo (diferencia entre el valor máximo 410-4 y el valor mínimo 410-3 de la entrada de tabla de correspondencia casi ordenada 410) será al menos tan amplio como debería ser, y posiblemente más. Un intervalo demasiado amplio significa que la extensión 300-2 puede explorarse sin necesidad para obtener la información solicitada. Esta posición conservadora es segura y es menos pesimista que invalidar las entradas de tabla de correspondencia casi ordenada 410 para extensiones 300-2 que tienen registros recién creados y no necesita la complejidad computacional ni la sobrecarga de almacenamiento de soportar múltiples versiones o deshacer cambios en entradas de tabla de correspondencia casi ordenada 410.

Cuando se actualiza un registro existente, si el espacio de información soporta múltiples versiones, una forma de realización supone de manera pesimista que sus valores de columna han ampliado lo máximo posible el intervalo de todas las entradas de tabla de correspondencia casi ordenada 410 asociadas a la extensión 300-2 que alberga la nueva versión del registro. Si el espacio de información no soporta múltiples versiones, una forma de realización supone de manera pesimista que sus valores de columna han ampliado lo máximo posible el intervalo de todas las entradas de tabla de correspondencia casi ordenada 410 asociadas a la extensión 300-2 que contiene el registro.

En algunos modos de funcionamiento, el espacio usado para almacenar información en un espacio de información 300-1 se reclama periódicamente. Por ejemplo, en un almacén de datos con una política de guardar los últimos 120 días de información de eventos, cualquier información de eventos que supere los 120 días puede archivarse. Las extensiones 300-2 usadas para almacenar esta información antigua pueden reclamarse y reutilizarse después para almacenar otra información.

En una forma de realización de la invención, la reclamación puede realizarse liberando los bloques 300-3 de las extensiones más bajas que están reclamándose y renumerando los bloques 300-3 de todas las extensiones 300-2 guardadas. Por ejemplo, supóngase que un espacio de información 300-1 consiste inicialmente en 10 extensiones 300-2 y que después de un periodo de tiempo la política dictaminó que la información almacenada en las 3 primeras extensiones 300-2 puede archivarse y que el espacio usado por esas 3 primeras extensiones 300-2 puede reclamarse. Si una extensión consiste en 24 bloques 300-3, entonces al primer bloque 300-3 de la cuarta extensión original 300-2 se le asigna el número 72 (3*24).

Sin embargo, después de la reclamación, el primer bloque 300-3 de la cuarta extensión original 300-2 debe renumerarse para convertirse en el primer bloque 300-3 de la primera extensión 300-2. Esto puede conseguirse restando a cada número de bloque guardado el número de bloques 300-3 reclamado, de modo que el nuevo número

de bloque del primer bloque 300-3 de la cuarta extensión original 300-2 pasaría a ser 0 (72 menos el número de bloques reclamado, que también es 72 en este ejemplo).

Cuando los bloques 300-3 de una extensión 300-2 se renumeran, su mapeo con el número de extensión puede cambiar de tal manera que invalide entradas de correspondencias casi ordenadas. En este caso puede usarse un procedimiento sencillo para restaurar la validez del establecimiento de correspondencias.

Si se ha reclamado un número N de extensiones 300-2 de bloques, entonces las entradas de tabla de correspondencia casi ordenada 410 para esas extensiones 300-2 se borran de la tabla de correspondencia casi ordenada 400. Además, el mismo número N debe restarse a todos los identificadores de extensión 410-5 de todas las entradas de tabla de correspondencia casi ordenada 410 restantes asociadas al espacio de información 300-1 que se ha reclamado.

El uso de correspondencias casi ordenadas con otras relaciones parcialmente ordenadas es similar al anterior. Un ejemplo es su uso con vistas materializadas ordenadas por sectores. En este ejemplo, las fases de ordenación son las habituales: Fase 1. Crear histograma e identificar límites de sector; Fase 2. Transferir datos a los sectores; Fase 3. Ordenar cada sector. Aunque este ejemplo es una ordenación de sectores sin mejoras, las correspondencias casi ordenadas también se aplican de manera apropiada a otras relaciones parcialmente ordenadas.

Normalmente, la Fase 1 es muy rápida y puede realizarse cuando se cargan datos. En la Fase 2 se leen los datos sin procesar, y los campos solicitados se distribuyen en sus sectores apropiados sin ordenarse. En la Fase 3 se ordenan cada uno de los sectores.

Con la forma de realización preferida de esta invención se crean y mantienen correspondencias casi ordenadas, como se ha descrito en las secciones anteriores, durante la Fase 2 y la Fase 3. Las correspondencias casi ordenadas creadas después de la Fase 2 son particularmente útiles ya que la relación parcialmente ordenada creada tras la Fase 2 puede usarse en consultas sin una exploración de tabla completa. En lo que respecta a consultas que tienen una cláusula de restricción o un orden según una cláusula en el índice de columna, solo se exploran aquellas partes del disco que pueden contener datos objetivo. Además, en lo que respecta a consultas que esperan datos perfectamente ordenados (datos de la Fase 3) procedentes del disco, estos datos de la Fase 2 pueden leerse, y una ordenación rápida de esos datos que satisfacen la restricción puede realizarse tras la lectura del disco.

Puesto que la Fase 1 y la Fase 2 requieren el 40% aproximadamente del tiempo total de ordenación de sector, el uso de una relación de Fase 2 permite que el tiempo de "ordenación" eficaz para una tabla sea el 40% de lo que sería en otro caso. Por tanto, la Fase 3 puede llevarse cabo sector a sector como una tarea en segundo plano.

Aunque la forma de realización preferida de esta invención se ha mostrado y descrito en particular con referencias a formas de realización preferidas de la misma, los expertos en la técnica deben entender que pueden realizarse varios cambios en la forma y los detalles sin apartarse del alcance de la forma de realización preferida de la invención definida por las reivindicaciones adjuntas.

REIVINDICACIONES

1. Un procedimiento implementado por ordenador para localizar datos deseados en una base de datos (300-1) usando una tabla de correspondencia casi ordenada (400), comprendiendo dicho procedimiento de manera secuencial:

dividir datos de la base de datos (300-1) en una pluralidad de extensiones (300-2);
generar estadísticas de correspondencias casi ordenadas de datos incluidos en cada extensión (300-2), donde las estadísticas de correspondencias casi ordenadas comprenden uno o más intervalos de datos incluidos en la extensión respectiva, donde cada uno de los intervalos consiste en un valor mínimo y un valor máximo de un atributo de los datos de la extensión respectiva, recopilando, para todas las extensiones, el valor mínimo y el valor máximo del atributo de los datos de la extensión respectiva,
anotar cada extensión (300-2) con la estadística de correspondencias casi ordenadas de los datos incluidos en la extensión respectiva,
donde los intervalos de datos asociados a la extensiones (300-2) están almacenados en la tabla de correspondencia casi ordenada;
seleccionar una o más extensiones (300-2) para las que los intervalos de datos asociados se solapan al menos parcialmente con un intervalo de datos de los datos deseados, en función de los valores mínimos y máximos de los intervalos de datos almacenados en la tabla de correspondencia casi ordenada, consultando la tabla de correspondencia casi ordenada para determinar un subconjunto de extensiones que pueden albergar los datos deseados; y
buscar la una o más extensiones seleccionadas para localizar los datos deseados.

2. El procedimiento según la reivindicación 1, en el que el intervalo de datos para cada extensión comprende uno o más intervalos de datos en función de uno o más atributos, y la etapa de seleccionar una o más extensiones comprende además:

seleccionar una o más extensiones (300-2) en función de un conjunto preseleccionado de atributos para los que intervalos de datos asociados se solapan al menos parcialmente con un intervalo de datos para los datos deseados, usando un conjunto preseleccionado de atributos de datos en función del uno o más atributos asociados a la extensión (300-2).

3. El procedimiento según la reivindicación 2, en el que el uno o más atributos están ordenados de manera lógica y la etapa de seleccionar una o más extensiones en función de un conjunto preseleccionado de atributos comprende además:

seleccionar una o más extensiones (300-2) en función de un conjunto preseleccionado de atributos en función de intervalos para esos atributos en un orden según un orden lógico dado de los atributos.

4. El procedimiento según la reivindicación 1, que comprende además: actualizar la anotación de extensión para una extensión a medida que se realizan cambios en datos almacenados en esa extensión (300-2).

5. El procedimiento según la reivindicación 1, en el que los tamaños de las extensiones no son iguales entre sí.

6. El procedimiento según la reivindicación 2, que comprende además: seleccionar un tamaño apropiado para una extensión particular en función de un tipo de atributo al que se refiere el intervalo de datos asociado.

7. El procedimiento según la reivindicación 1, en el que los tamaños de las extensiones (300-2) son iguales entre sí.

8. El procedimiento según la reivindicación 7, que comprende además: seleccionar dinámicamente un tamaño para las extensiones (300-2) en función de cambios en el tamaño de la base de datos.

9. El procedimiento según la reivindicación 1, en el que la base de datos es una vista materializada de una base de datos.

10. El procedimiento según la reivindicación 1, en el que una anotación en un intervalo de datos para una extensión (300-2) está coubicada físicamente con esa extensión.

11. El procedimiento según la reivindicación 1, en el que la una o más extensiones seleccionadas (300-2) comprenden subextensiones anotadas, y la etapa de buscar la una o más extensiones seleccionadas (300-2) comprende además:

seleccionar una o más subextensiones para las que los intervalos de datos asociados pueden contener los datos deseados; y
buscar la una o más subextensiones seleccionadas para localizar los datos deseados.

12. El procedimiento según la reivindicación 11, en el que los intervalos de datos asociados a las subextensiones están basados en uno o más atributos diferentes a los atributos asociados a intervalos de datos asociados a las extensiones (300-2).

13. El procedimiento según la reivindicación 1, en el que la base de datos es una base de datos distribuida almacenada en uno o más nodos.

14. El procedimiento según la reivindicación 1, que comprende además:

determinar si establecer al menos una tabla de correspondencia casi ordenada para un espacio de información dentro de la base de datos.

15. El procedimiento según la reivindicación 14, en el que la etapa de determinar si establecer tablas de correspondencias casi ordenadas para el espacio de información particular comprende además:

identificar al menos un índice de columna del espacio de información como candidato a tabla de correspondencia casi ordenada.

16. El procedimiento según la reivindicación 15, en el que la etapa de identificar al menos un índice de columna como candidato a tabla de correspondencia casi ordenada comprende además:

identificar al menos un índice de columna candidato en función de un tipo de datos de columna asociado.

17. El procedimiento según la reivindicación 14, que comprende además:

crear correspondencias casi ordenadas para todos los índices de columna candidatos identificados.

18. El procedimiento según la reivindicación 14, en el que la etapa de determinar si establecer tablas de correspondencias casi ordenadas para el espacio de información particular comprende además:

comparar intervalos de datos de extensiones del espacio de información; y determinar si una cantidad de solapamiento entre intervalos de datos de las extensiones del espacio de información es lo bastante pequeña como para indicar la necesidad de una tabla de correspondencia casi ordenada.

19. El procedimiento según la reivindicación 14, en el que la etapa de determinar si establecer tablas de correspondencias casi ordenadas para el espacio de información particular comprende además:

determinar si el espacio de información es lo bastante grande como para indicar la necesidad de una correspondencia casi ordenada.

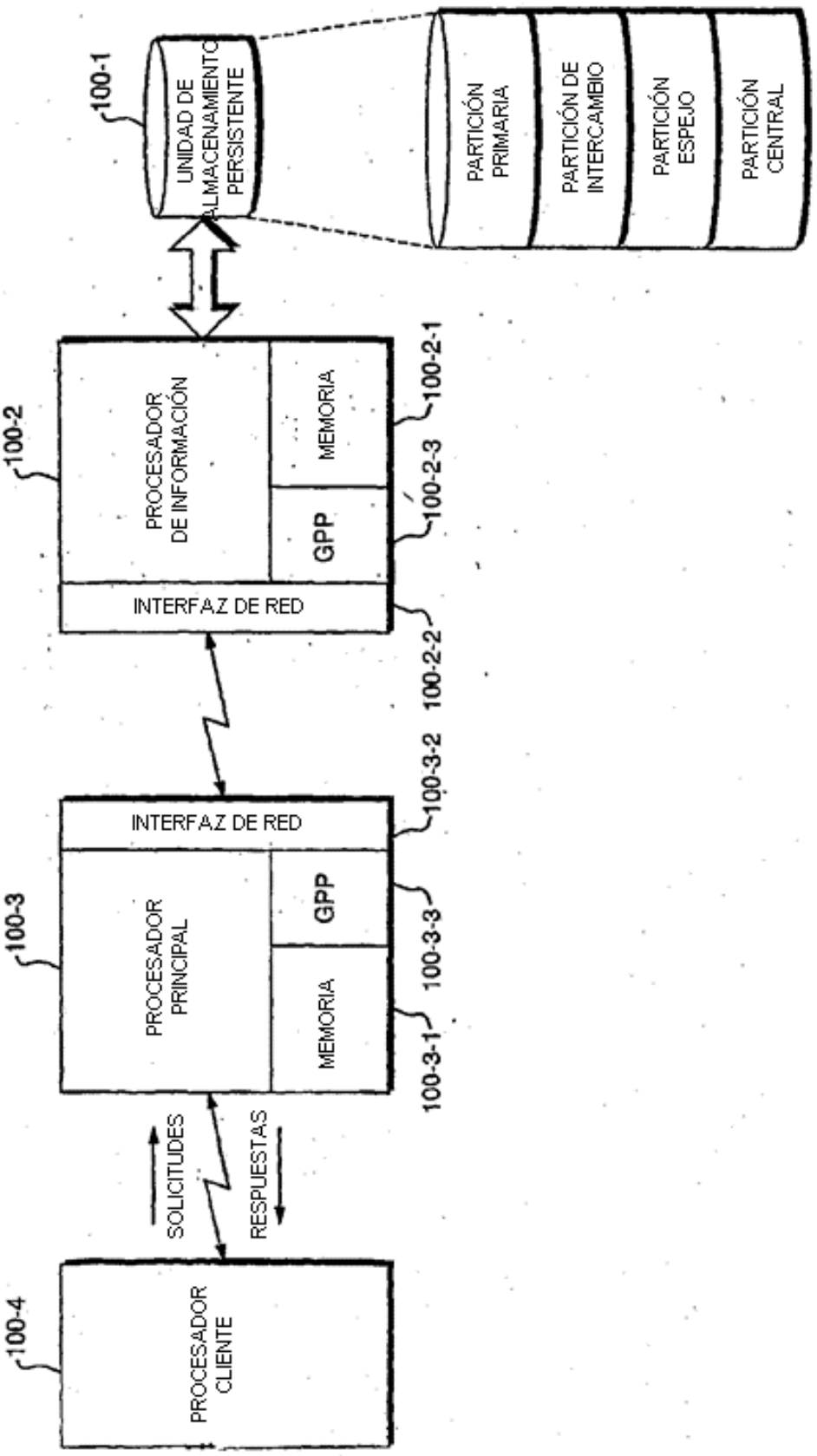

100-1
UNIDAD DE ALMACENAMIENTO PERSISTENTE
PARTICIÓN PRIMARIA
PARTICIÓN DE INTERCAMBIO
PARTICIÓN ESPEJO
PARTICIÓN CENTRAL
100-2
PROCESADOR DE INFORMACIÓN
MEMORIA
GPP
INTERFAZ DE RED
100-2-1
100-2-3
100-2-2
100-3-2
100-3
INTERFAZ DE RED
PROCESADOR PRINCIPAL
GPP
MEMORIA
100-3-3
100-3-1
SOLICITUDES
RESPUESTAS
100-4
PROCESADOR CLIENTE

FIG. 1

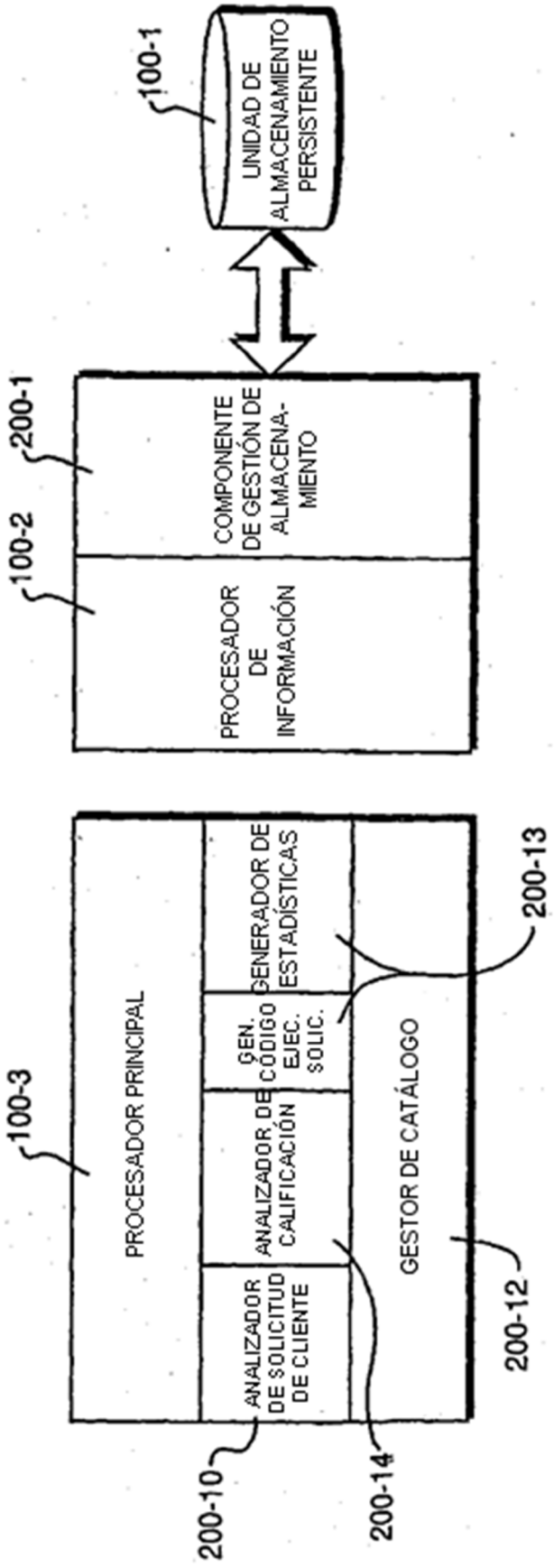

100-1
UNIDAD DE ALMACENAMIENTO PERSISTENTE
200-1
COMPONENTE DE GESTIÓN DE ALMACENA-MIENTO
100-2
PROCESADOR DE INFORMACIÓN
100-3
PROCESADOR PRINCIPAL
ANALIZADOR DE SOLICITUD DE CLIENTE
ANALIZADOR DE CALIFICACIÓN
GEN. CÓDIGO EJEC. SOLIC.
GENERADOR DE ESTADÍSTICAS
GESTOR DE CATÁLOGO
200-10
200-14
200-12
200-13

FIG. 2

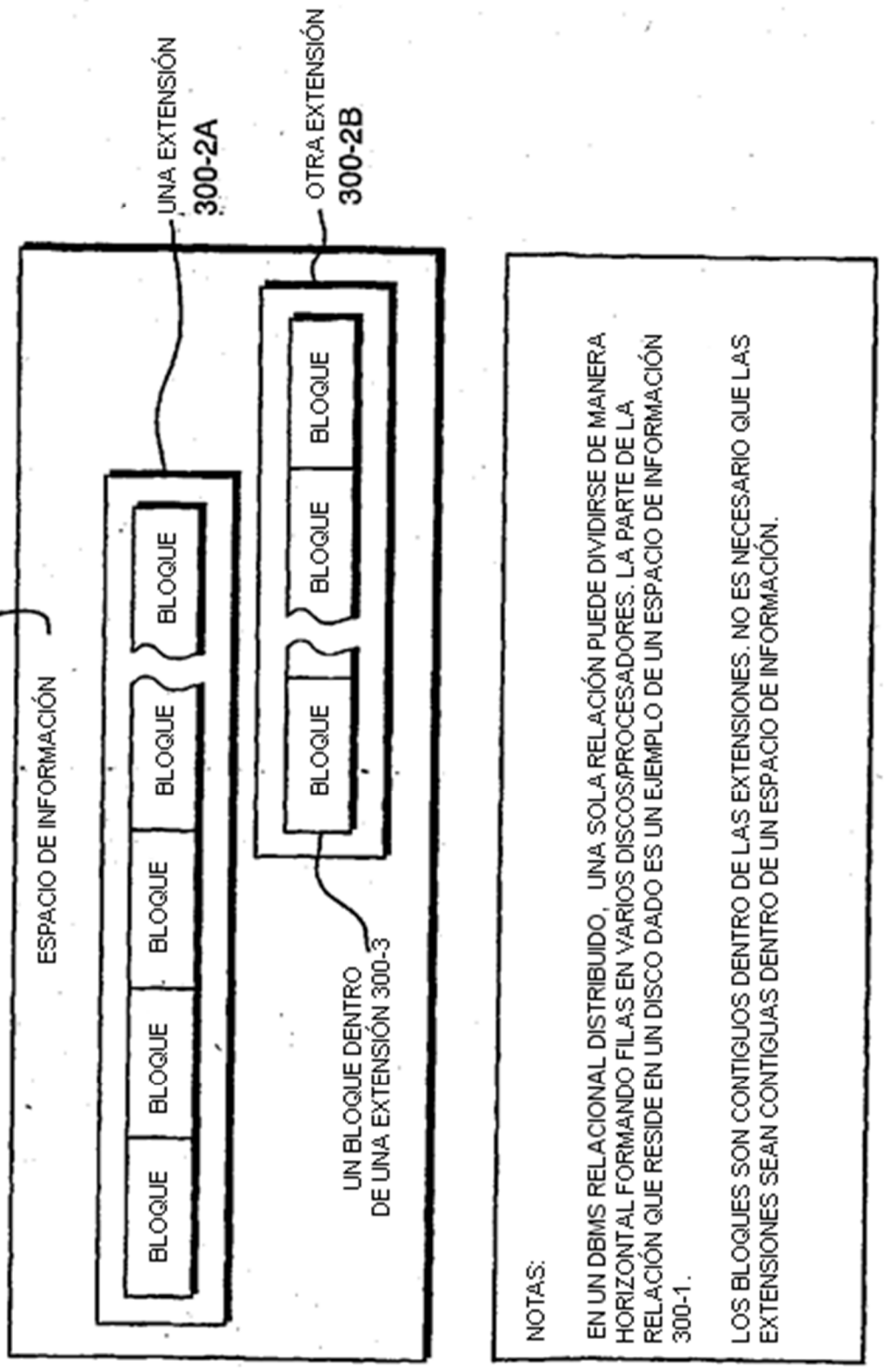

300-1
ESPACIO DE INFORMACIÓN
UNA EXTENSIÓN
300-2A
OTRA EXTENSIÓN
300-2B
BLOQUE
BLOQUE
BLOQUE
BLOQUE
BLOQUE
BLOQUE
BLOQUE
BLOQUE
UN BLOQUE DENTRO
DE UNA EXTENSIÓN 300-3
NOTAS:
EN UN DBMS RELACIONAL DISTRIBUIDO, UNA SOLA RELACIÓN PUEDE DIVIDIRSE DE MANERA HORIZONTAL FORMANDO FILAS EN VARIOS DISCOS/PROCESADORES. LA PARTE DE LA RELACIÓN QUE RESIDE EN UN DISCO DADO ES UN EJEMPLO DE UN ESPACIO DE INFORMACIÓN 300-1.
LOS BLOQUES SON CONTIGUOS DENTRO DE LAS EXTENSIONES. NO ES NECESARIO QUE LAS EXTENSIONES SEAN CONTIGUAS DENTRO DE UN ESPACIO DE INFORMACIÓN.

FIG. 3

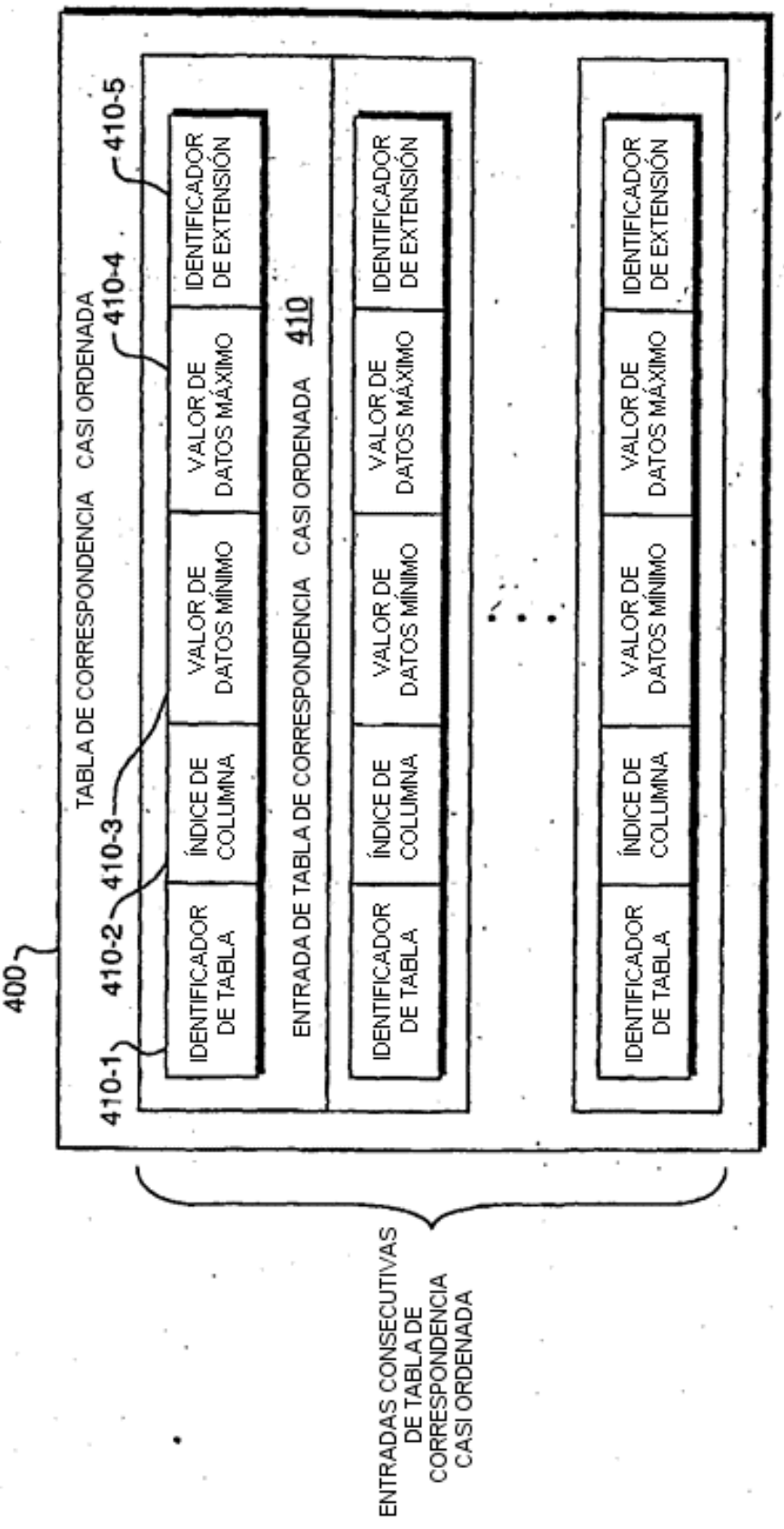

400
TABLA DE CORRESPONDENCIA CASI ORDENADA
410-1
410-2
410-3
410-4
410-5
ENTRADA DE TABLA DE CORRESPONDENCIA CASI ORDENADA 410
IDENTIFICADOR DE TABLA
ÍNDICE DE COLUMNA
VALOR DE DATOS MÍNIMO
VALOR DE DATOS MÁXIMO
IDENTIFICADOR DE EXTENSIÓN
ENTRADAS CONSECUTIVAS DE TABLA DE CORRESPONDENCIA CASI ORDENADA

FIG. 4

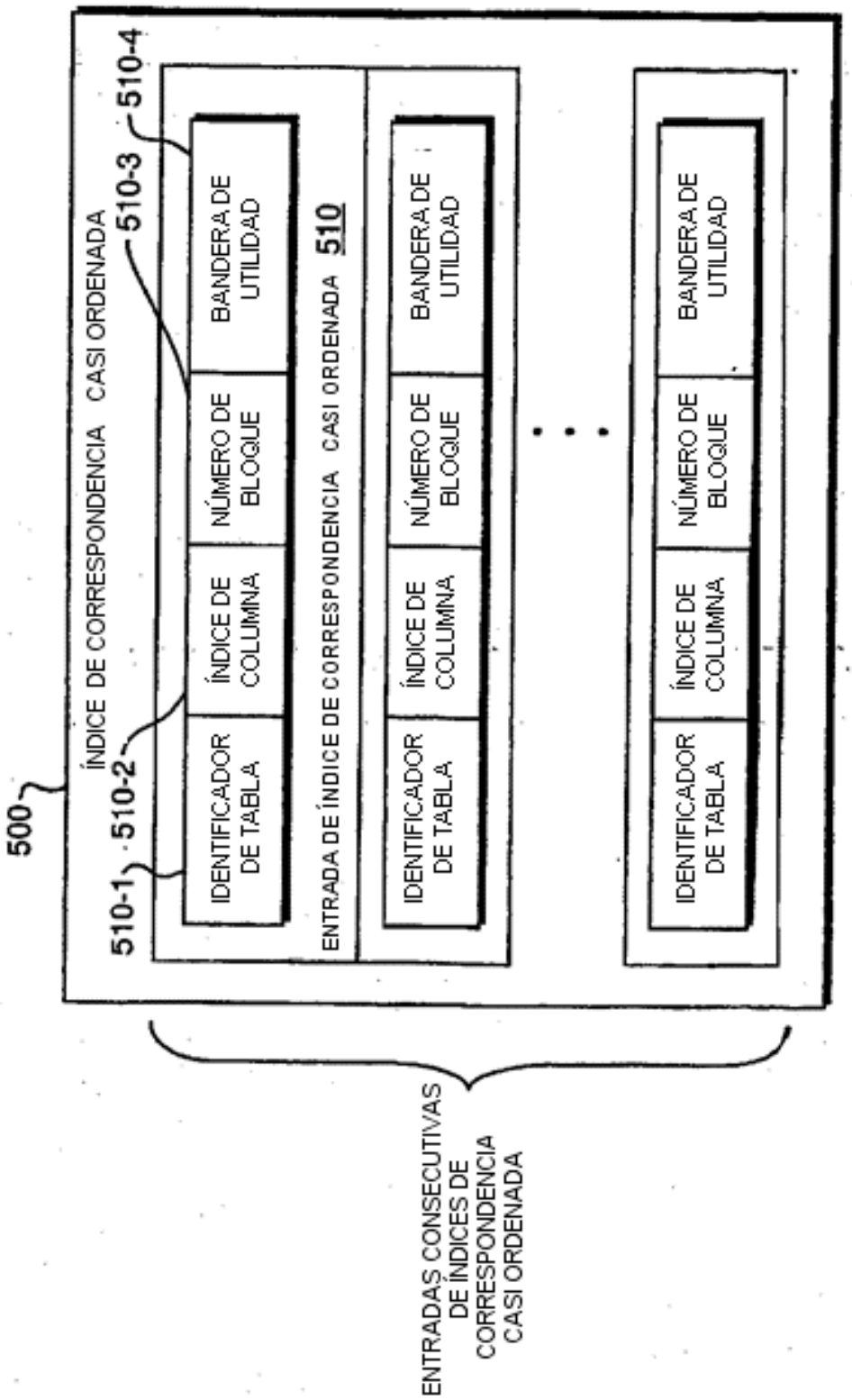

500
ÍNDICE DE CORRESPONDENCIA CASI ORDENADA
510-1
510-2
510-3
510-4
IDENTIFICADOR DE TABLA
ÍNDICE DE COLUMNA
NÚMERO DE BLOQUE
BANDERA DE UTILIDAD
ENTRADA DE ÍNDICE DE CORRESPONDENCIA CASI ORDENADA 510
ENTRADAS CONSECUTIVAS DE ÍNDICES DE CORRESPONDENCIA CASI ORDENADA

FIG. 5

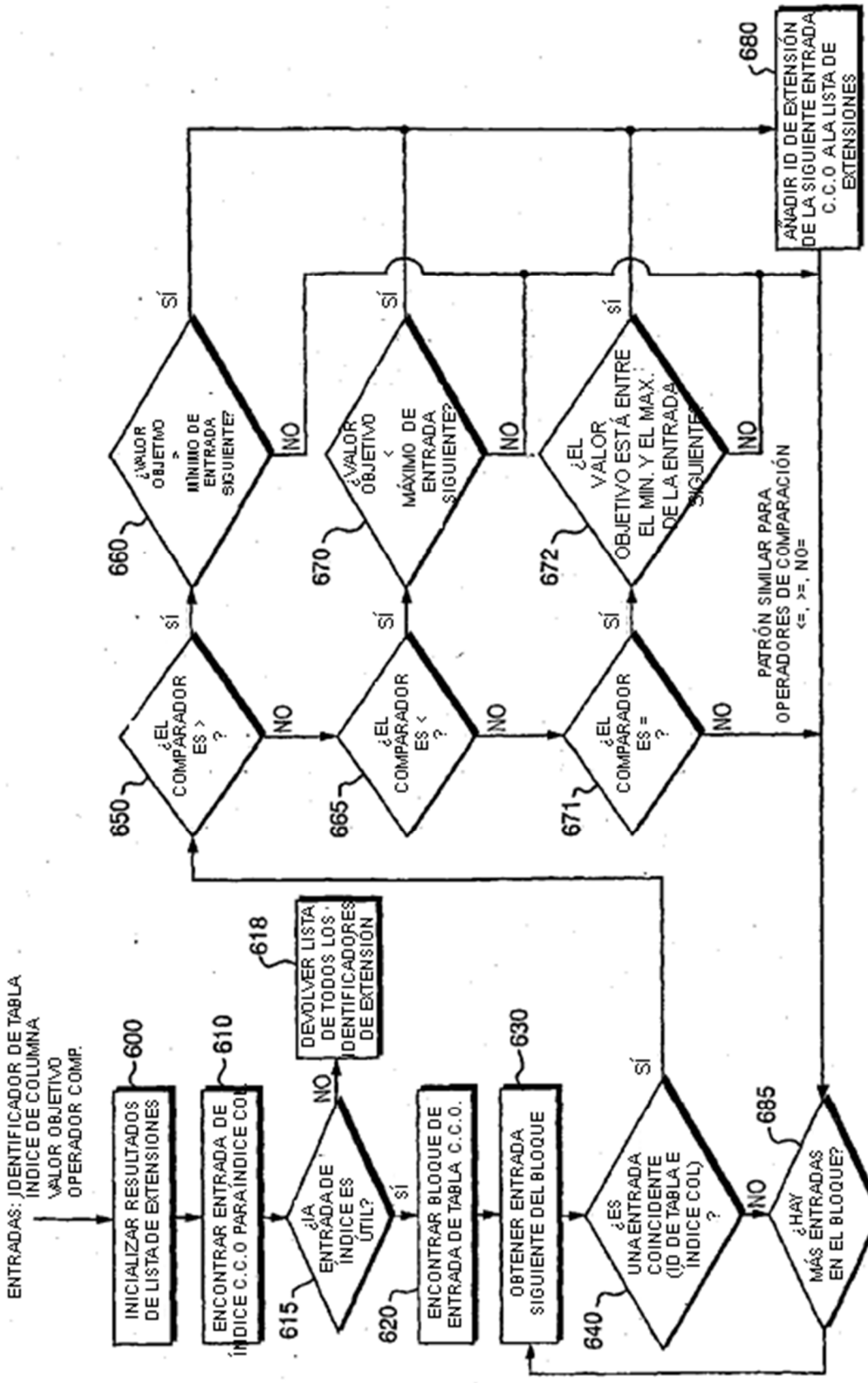
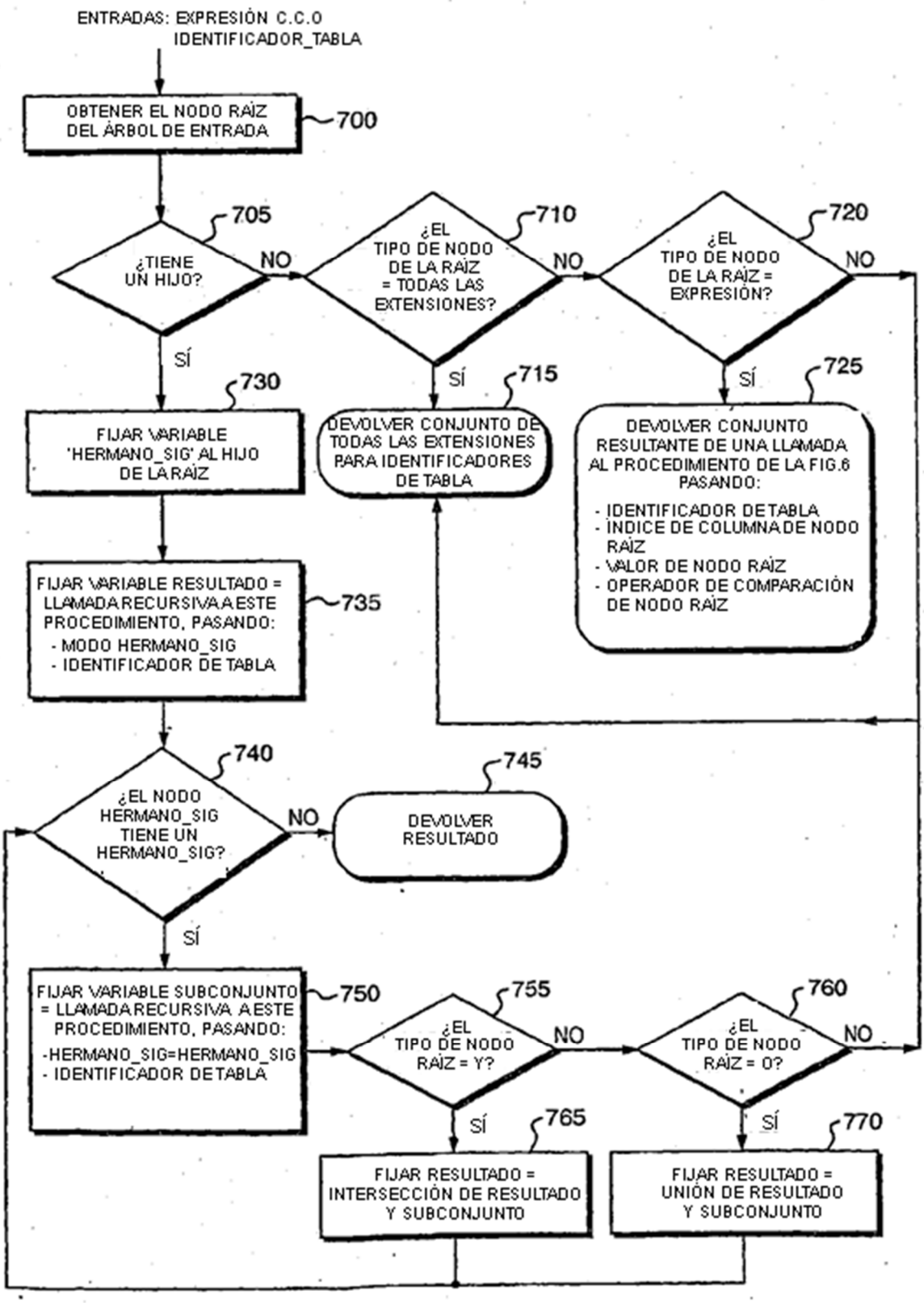
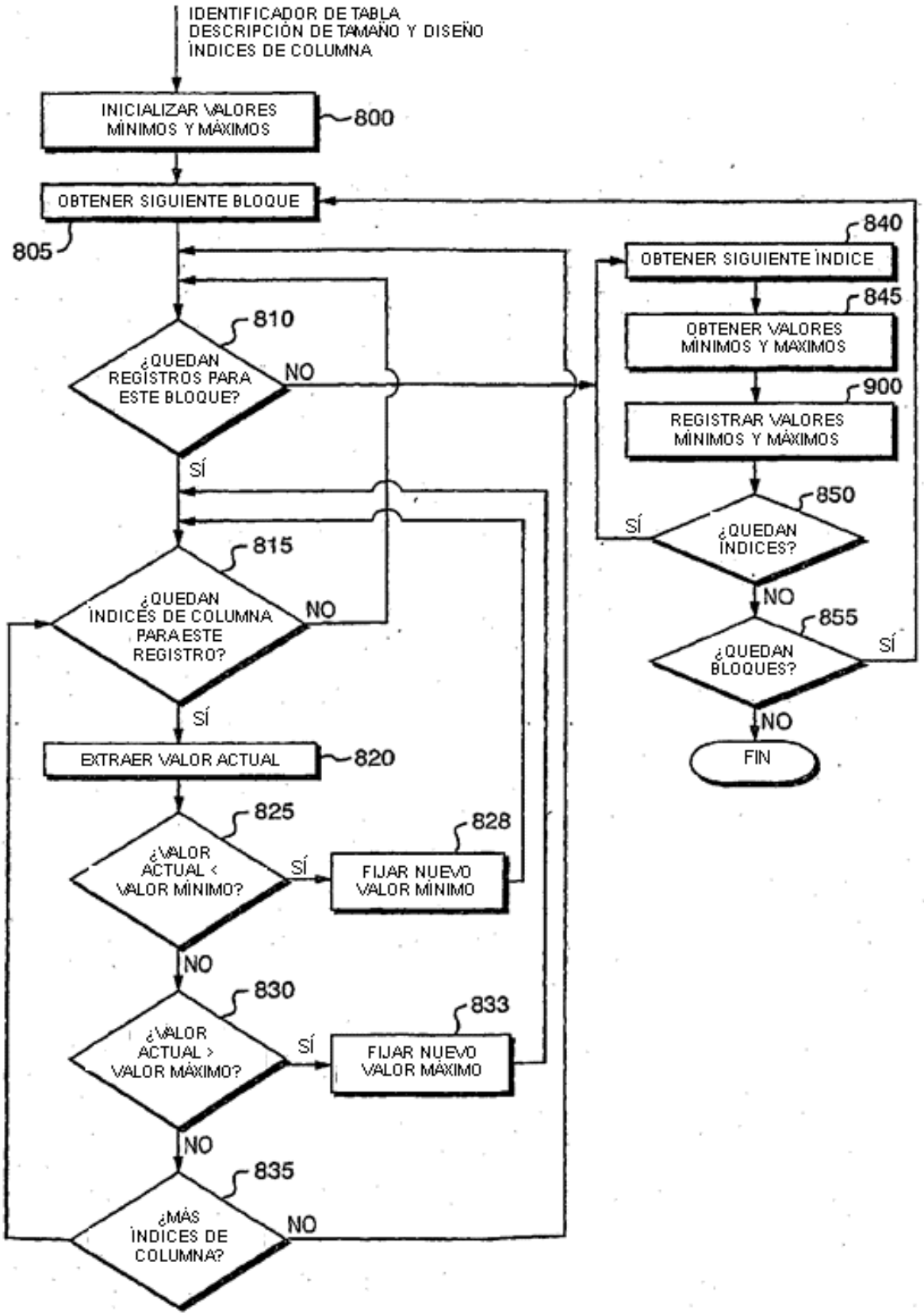

ENTRADAS: IDENTIFICADOR DE TABLA
ÍNDICE DE COLUMNA
VALOR OBJETIVO
OPERADOR COMP.
600
INICIALIZAR RESULTADOS DE LISTA DE EXTENSIONES
610
ENCONTRAR ENTRADA DE ÍNDICE C.C.O PARA ÍNDICE COL
615
¿LA ENTRADA DE ÍNDICE ES ÚTIL?
NO
618
DEVOLVER LISTA DE TODOS LOS IDENTIFICADORES DE EXTENSIÓN
SÍ
620
ENCONTRAR BLOQUE DE ENTRADA DE TABLA C.C.O.
630
OBTENER ENTRADA SIGUIENTE DEL BLOQUE
640
¿ES UNA ENTRADA COINCIDENTE (ID DE TABLA E ÍNDICE COL) ?
SÍ
680
AÑADIR ID DE EXTENSIÓN DE LA SIGUIENTE ENTRADA C.C.O A LA LISTA DE EXTENSIONES
NO
685
¿HAY MÁS ENTRADAS EN EL BLOQUE?
NO
690
DEVOLVER LISTA DE ID DE EXTENSIÓN
650
¿EL COMPARADOR ES > ?
SÍ
660
¿VALOR OBJETIVO > MÍNIMO DE ENTRADA SIGUIENTE?
NO
665
¿EL COMPARADOR ES < ?
SÍ
670
¿VALOR OBJETIVO < MÁXIMO DE ENTRADA SIGUIENTE?
NO
671
¿EL COMPARADOR ES = ?
SÍ
672
¿EL VALOR OBJETIVO ESTÁ ENTRE EL MIN. Y EL MAX. DE LA ENTRADA SIGUIENTE?
NO
PATRÓN SIMILAR PARA OPERADORES DE COMPARACIÓN <=, >=, NO=

FIG. 6


ENTRADAS: EXPRESIÓN C.C.O
IDENTIFICADOR_TABLA
OBTENER EL NODO RAÍZ DEL ÁRBOL DE ENTRADA
700
¿TIENE UN HIJO?
705
NO
SÍ
¿EL TIPO DE NODO DE LA RAÍZ = TODAS LAS EXTENSIONES?
710
NO
SÍ
¿EL TIPO DE NODO DE LA RAÍZ = EXPRESIÓN?
720
NO
SÍ
FIJAR VARIABLE 'HERMANO_SIG' AL HIJO DE LA RAÍZ
730
DEVOLVER CONJUNTO DE TODAS LAS EXTENSIONES PARA IDENTIFICADORES DE TABLA
715
DEVOLVER CONJUNTO RESULTANTE DE UNA LLAMADA AL PROCEDIMIENTO DE LA FIG.6 PASANDO:
- IDENTIFICADOR DE TABLA
- ÍNDICE DE COLUMNA DE NODO RAÍZ
- VALOR DE NODO RAÍZ
- OPERADOR DE COMPARACIÓN DE NODO RAÍZ
725
FIJAR VARIABLE RESULTADO = LLAMADA RECURSIVA A ESTE PROCEDIMIENTO, PASANDO:
- MODO HERMANO_SIG
- IDENTIFICADOR DE TABLA
735
¿EL NODO HERMANO_SIG TIENE UN HERMANO_SIG?
740
NO
SÍ
DEVOLVER RESULTADO
745
FIJAR VARIABLE SUBCONJUNTO = LLAMADA RECURSIVA A ESTE PROCEDIMIENTO, PASANDO:
-HERMANO_SIG=HERMANO_SIG
- IDENTIFICADOR DE TABLA
750
¿EL TIPO DE NODO RAÍZ = Y?
755
NO
SÍ
¿EL TIPO DE NODO RAÍZ = O?
760
NO
SÍ
FIJAR RESULTADO = INTERSECCIÓN DE RESULTADO Y SUBCONJUNTO
765
FIJAR RESULTADO = UNIÓN DE RESULTADO Y SUBCONJUNTO
770


IDENTIFICADOR DE TABLA
DESCRIPCIÓN DE TAMAÑO Y DISEÑO
ÍNDICES DE COLUMNA
INICIALIZAR VALORES MÍNIMOS Y MÁXIMOS
800
OBTENER SIGUIENTE BLOQUE
805
810
¿QUEDAN REGISTROS PARA ESTE BLOQUE?
NO
SÍ
815
¿QUEDAN ÍNDICES DE COLUMNA PARA ESTE REGISTRO?
NO
SÍ
EXTRAER VALOR ACTUAL
820
825
¿VALOR ACTUAL < VALOR MÍNIMO?
SÍ
828
FIJAR NUEVO VALOR MÍNIMO
NO
830
¿VALOR ACTUAL > VALOR MÁXIMO?
SÍ
833
FIJAR NUEVO VALOR MÁXIMO
NO
835
¿MÁS ÍNDICES DE COLUMNA?
NO
840
OBTENER SIGUIENTE ÍNDICE
845
OBTENER VALORES MÍNIMOS Y MÁXIMOS
900
REGISTRAR VALORES MÍNIMOS Y MÁXIMOS
850
¿QUEDAN ÍNDICES?
SÍ
NO
855
¿QUEDAN BLOQUES?
SÍ
NO
FIN

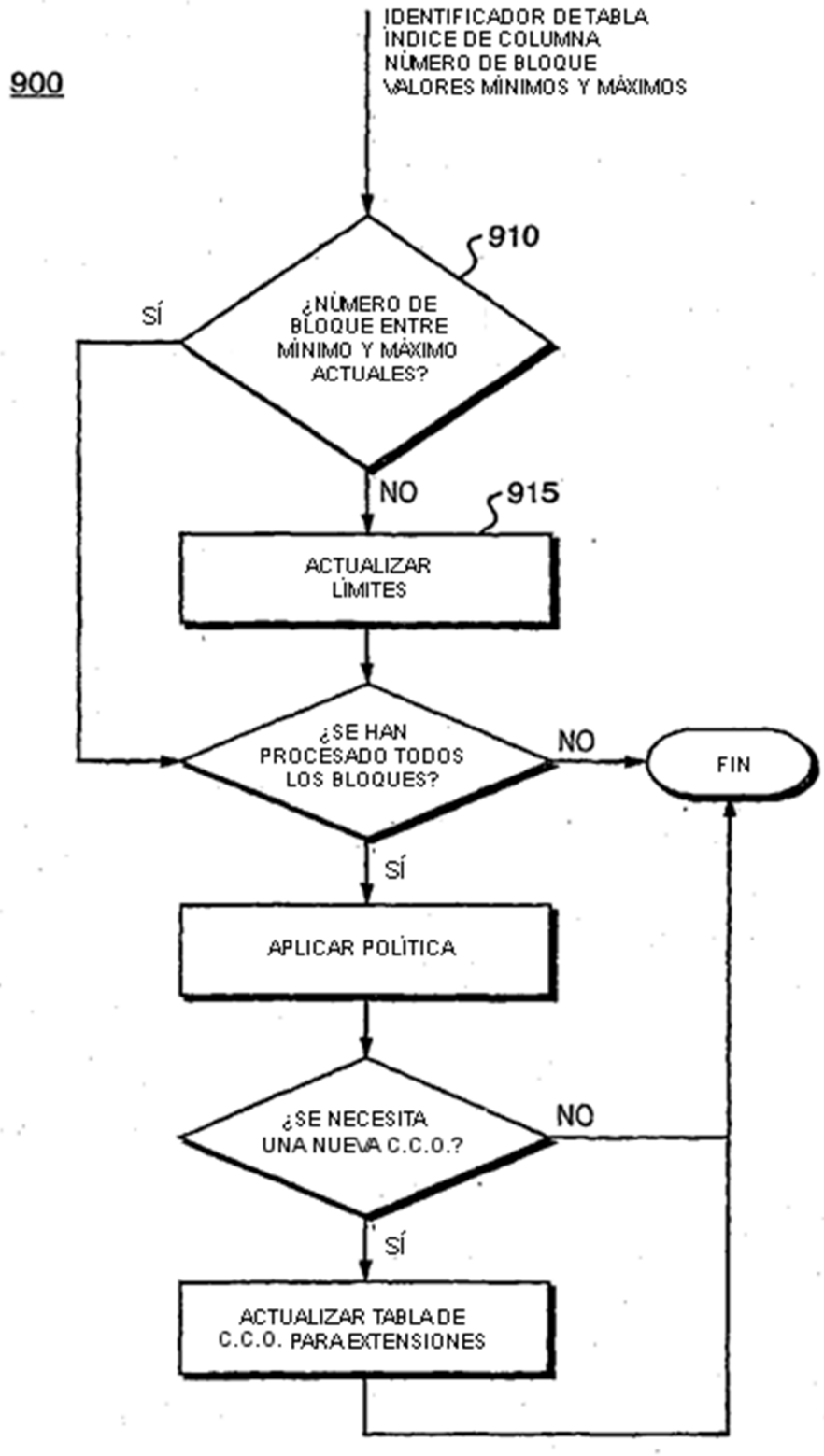

900
IDENTIFICADOR DE TABLA
ÍNDICE DE COLUMNA
NÚMERO DE BLOQUE
VALORES MÍNIMOS Y MÁXIMOS
910
¿NÚMERO DE BLOQUE ENTRE MÍNIMO Y MÁXIMO ACTUALES?
SÍ
NO
915
ACTUALIZAR LÍMITES
¿SE HAN PROCESADO TODOS LOS BLOQUES?
NO
FIN
SÍ
APLICAR POLÍTICA
¿SE NECESITA UNA NUEVA C.C.O.?
NO
SÍ
ACTUALIZAR TABLA DE C.C.O. PARA EXTENSIONES

FIG. 9

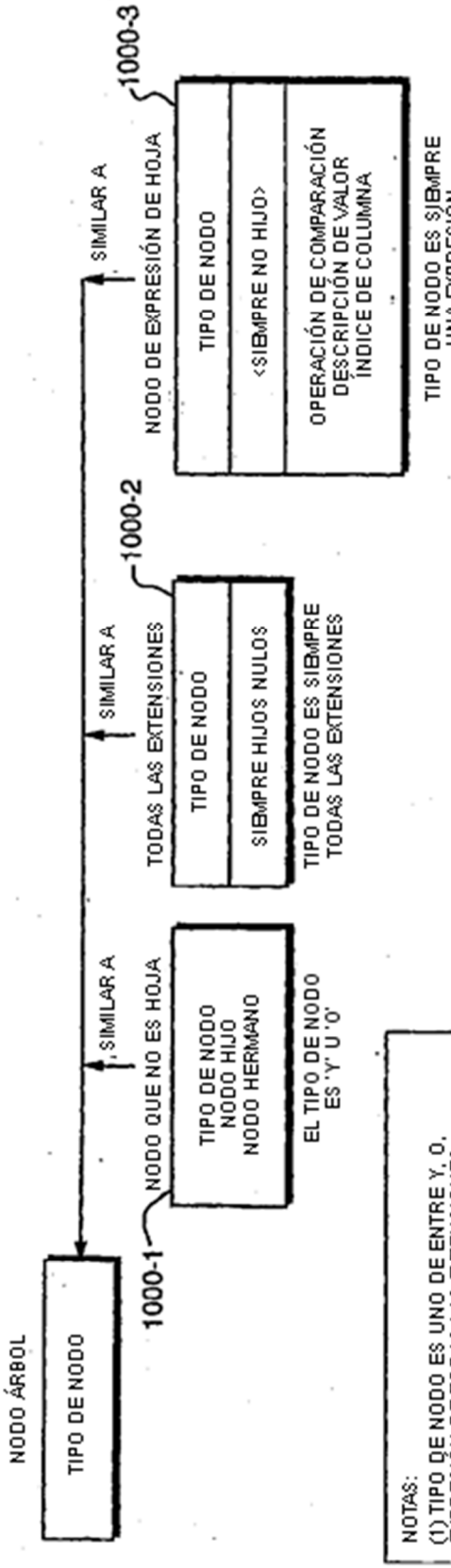


FIG. 10A

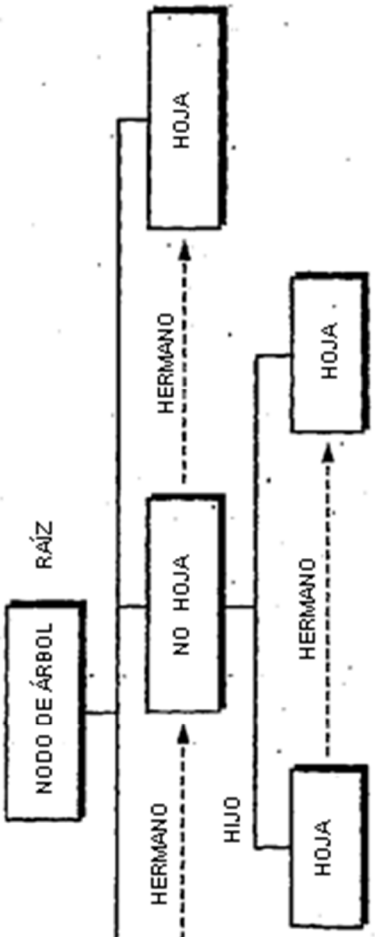


FIG. 10B