

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 612 260**

51 Int. Cl.:

**G06F 17/30** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **21.09.2012 PCT/FI2012/050911**

87 Fecha y número de publicación internacional: **28.03.2013 WO13041774**

96 Fecha de presentación y número de la solicitud europea: **21.09.2012 E 12834563 (4)**

97 Fecha y número de publicación de la concesión europea: **02.11.2016 EP 2758901**

54 Título: **Mecanismo para actualizaciones en un motor de base de datos**

30 Prioridad:

**22.09.2011 FI 20115932**

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

**16.05.2017**

73 Titular/es:

**RETAIL LOGISTICS EXCELLENCE - RELEX OY  
(100.0%)**

**Postintaival 7  
00230 Helsinki, FI**

72 Inventor/es:

**NIKULA, MARKO**

74 Agente/Representante:

**DE ELZABURU MÁRQUEZ, Alberto**

**ES 2 612 260 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**

Mecanismo para actualizaciones en un motor de base de datos

**Campo de la invención**

5 La presente invención está relacionada generalmente con lo que comúnmente se llama un motor de base de datos, esto es, un aparato programado de procesamiento de datos que implementa operaciones internas de una base de datos, tales como creación, lectura, actualización y borrado de datos, independientemente del significado cognitivo de los datos. Más particularmente, la invención está relacionada con las prestaciones de una base de datos en presencia de un gran volumen de actualizaciones.

**Antecedentes de la invención**

10 El documento de referencia 1 (Stonebraker et al.) explica generalmente cómo difieren bases de datos orientadas por columnas de las bases de datos más tradicionales orientadas por filas. En términos muy generales, en un motor tradicional orientado por filas de base de datos, en el almacenamiento se colocan contiguamente atributos de un registro (o tupla). Motores orientados por filas de base de datos proporcionan buenas prestaciones para escritura (insertar y actualizar datos), y son eficientes para Procesamiento de Transacciones En Línea ("OLTP"). Stonebraker  
15 et al. también describe un ejemplo de un motor orientado por columnas de base de datos en el que los valores para cada única columna (o atributo) se almacenan contiguos. Una disposición de almacenamiento orientada por columnas supuestamente permite mejor compresión de datos, porque típicamente la variación entre el mismo atributo para todos los registros es menor que la variación entre todos los atributos de un único registro. En un servidor de base de datos moderno el cuello de botella de las prestaciones es a menudo la velocidad de transferencia de datos hacia y desde el almacenamiento permanente, y una mejor compresión alivia este cuello de botella. Como resultado, los motores orientados por columnas de base de datos son cada vez más populares para aplicaciones que implican leer grandes volúmenes de datos. Dichas aplicaciones se denominan frecuentemente aplicaciones de Procesamiento Analítico En Línea ("OLAP"). Los motores de base de datos OLAP se optimizan para leer o procesamiento de consultas en aplicaciones en donde una consulta puede expandirse a una gran cantidad de  
20 todos los datos almacenados en la base de datos.

Uno de los problemas asociados con la disposición anterior está relacionado con el hecho de que cuando se va a actualizar un atributo de un registro en la base de datos, puede no ser posible actualizar esa parte de datos en el sitio en la forma comprimida. Típicamente, se tiene que volver a comprimir el bloque entero de datos comprimidos. Si hay un gran volumen de actualizaciones a procesar, esta recompresión puede tener un gran efecto adverso en las  
30 prestaciones de actualización en una base de datos orientada por columnas que utiliza compresión. El documento US 2008/059492 describe sistemas y métodos para acceso agrupado a tantas columnas como sea posible dada una mezcla particular de consulta en marcha y una cantidad restringida de espacio en disco. Una base de datos comprimida se divide en grupo de columnas, cada columna tiene duplicados eliminados y clasificados. Luego ciertos grupos se transfieren a una memoria rápida dependiendo del registro de consultas recibidas previamente. El documento US 2005/050083 describe una técnica para procesar datos de entrada. Se reciben múltiples filas de entrada para cargarse en una primera estructura. Cada fila de entrada de las múltiples filas de entrada se procesa para clasificar cada fila de entrada como una de una fila de inserción y una fila de actualización, en donde los duplicados de entrada se almacenan en la primera estructura y entradas de índice para los duplicados de entrada se almacenan en una segunda estructura. Después de que se hayan procesado las múltiples filas de entrada, los duplicados de entrada se reaplican automáticamente a la primera estructura y se procesan las entradas de índice almacenadas en la segunda estructura. El documento US 2010/235335 describe un sistema informático de base de datos de almacenamiento por columnas que responde a peticiones de base de datos para la actualización y recuperación de datos desde dentro de una tabla estable de datos que permite el almacenamiento de tuplas de base de datos dentro de una estructura organizada de base de datos de almacenamiento por columnas. Se implementa una estructura de datos en árbol delta posicional en el espacio de memoria del sistema informático y se acopla funcionalmente en un recorrido de transferencia de datos de actualización entre una interfaz de motor de base de datos y la tabla estable de datos. La estructura de datos en árbol delta posicional incluye una capa de almacenamiento de datos diferencial operativa para almacenar valores de datos de actualización diferencial en referencia relativa posicionalmente definida con tuplas de base de datos almacenadas por la tabla estable de datos.

**50 Descripción de la invención**

Un objeto de la presente invención es así proporcionar un método, un aparato y un producto de programa informático para mitigar uno o más de los problemas identificados anteriormente. El objeto de la invención se logra mediante aspectos de las invenciones como se define en las reivindicaciones independientes adjuntas. Las reivindicaciones dependientes y la siguiente descripción detallada y dibujos se relacionan con realizaciones específicas que resuelven problemas adicionales y/o proporcionan beneficios adicionales.

Un aspecto de la invención es un método según la reivindicación 1. Otros aspectos de la invención incluyen un aparato para llevar a cabo el método y un producto de programa informático cuya ejecución en un sistema de

procesamiento de datos programado apropiadamente configurado provoca que el sistema lleve a cabo el método según la reivindicación 1.

Otro aspecto de la invención incluye un sistema servidor de base de datos configurado para llevar a cabo el método inventivo. Incluso otro aspecto de la invención es un soporte de programa informático que incorpora un paquete de programas informáticos para llevar a cabo el método inventivo cuando se ejecuta el paquete de programas en el sistema servidor de base de datos.

La invención se puede implementar como método para manejar una base de datos implementada en ordenador. Como bien se sabe, el funcionamiento de una base de datos informatizada comprende varias fases, tales como fase de preparación de base de datos, una fase de escritura, una fase de actualización, una fase de consulta y una fase de confirmación. En un método según la invención, un servidor de base de datos realiza las siguientes etapas para cada una de una o más columnas en cada una de una o más tablas:

- La fase de preparación comprende asignar al menos una estructura primaria de datos, al menos una estructura secundaria de datos y al menos una estructura terciaria de datos. Para los propósitos de la presente invención, la fase de preparación es la fase en donde se asignan diversas estructuras de datos. La asignación de estructuras de datos puede ser fija o dinámica, y preferiblemente se implementa una estrategia de asignación mezclada, de manera que la asignación es fija para la estructura(s) primaria(s) de datos pero dinámica para las estructuras secundaria y terciaria de datos. Asignación dinámica significa que las estructuras de datos pueden crecer y contraerse según sea necesario.

Para una tabla no particionada, cada una de la estructura primaria, secundaria y terciaria de datos es suficiente por columna, pero para tablas particionadas horizontalmente, se debe asignar una estructura primaria, una secundaria y una terciaria separadas de datos para cada partición horizontal de una columna. Por motivos de brevedad y claridad, las etapas restantes del método se describirán para el caso no particionado.

- La fase de escritura comprende escribir contenido en columnas en un formato comprimido en la estructura primaria de datos. Esta etapa se puede realizar de manera similar a escribir contenido en columnas en una base de datos conocida.

- La fase de actualización comprende escribir un valor actualizado para cada columna actualizada de cada fila actualizada en la estructura terciaria de datos de la columna respectiva, junto con una clave de almacenamiento que indica una fila correspondiente en la estructura primaria de datos. Así se escriben dos elementos para cada valor actualizado, es decir el propio valor y una clave de almacenamiento que vincula el valor con su fila correspondiente en la estructura primaria de datos. La estructura primaria de datos no tiene que tener una clave de almacenamiento explícita, y se puede usar el número de fila como clave de almacenamiento implícita. Pero como la estructura terciaria de datos almacena únicamente valores actualizados en el orden en el que se han actualizado los valores, las filas de la estructura terciaria de datos no tienen una correspondencia uno a uno con las filas de la estructura primaria de datos. Por tanto existe la necesidad de una clave de almacenamiento explícita que asocie cada valor actualizado en la estructura terciaria de datos con su homólogo en la estructura primaria de datos.

- La fase de consulta comprende leer los valores actualizados y las claves de almacenamiento de la estructura terciaria de datos, clasificar los valores actualizados por sus claves de almacenamiento y almacenar los valores clasificados y las claves de almacenamiento en el orden clasificado en la estructura secundaria de datos, y limpiar la estructura terciaria de datos. Como resultado de esta etapa, tanto la estructura primaria de datos como la estructura secundaria se clasifican según la misma clave, es decir la clave de almacenamiento que asocia cada valor de la estructura secundaria de datos con su homólogo (valor anterior a la actualización) en la estructura primaria de datos. Entonces el ordenador servidor responde a la consulta de base de datos preguntando a la estructura primaria de datos y a la estructura secundaria de datos. Esto significa que el valor para una clave de almacenamiento se lee de la estructura secundaria de datos si la estructura secundaria de datos contiene un valor para esa clave de almacenamiento. De lo contrario el valor se lee de la estructura primaria de datos.

- La fase de confirmación de base de datos comprende producir una versión nueva de la estructura primaria de datos combinando la estructura secundaria de datos y una versión antigua de la estructura primaria de datos. De manera similar a la operación de preguntar, la operación de combinación comprende leer un valor para una clave de almacenamiento de la estructura secundaria de datos si el valor existe o de lo contrario de la estructura primaria de datos.

**Breve descripción de los dibujos**

A continuación, la invención se describirá con mayor detalle por medio de realizaciones específicas con referencia a los dibujos adjuntos, en los que

La figura 1 ilustra esquemáticamente un principio de funcionamiento de la invención;

La figura 2 muestra esquemáticamente un diagrama de bloques de un sistema servidor de base de datos;

La figura 3 explica un principio de funcionamiento de la invención; y

La figura 4 ilustra aislamiento mutuo de transacciones concurrentes.

## 5 Descripción detallada de las realizaciones específicas

La figura 1 ilustra esquemáticamente un principio de funcionamiento de la invención. En aras de claridad y brevedad, la figura 1 únicamente ilustra una estructura de datos para una columna de una tabla de una base de datos en columnas, y cualesquiera columnas y tablas restantes se pueden procesar de manera similar. Una base de datos en columnas, indicada generalmente por el numeral de referencia 100, comprende una estructura primaria de datos 110, una estructura secundaria de datos 120 y una estructura terciaria de datos 130. Inicialmente, cuando la base de datos está vacía, se escriben datos nuevos en la estructura primaria de datos 110. Los números 1 a 5 representan esquemáticamente claves de almacenamiento dentro de la estructura primaria de datos 110. Después, cuando se va a escribir contenido actualizado en la base de datos 100, el contenido actualizado se escribe en la estructura terciaria de datos 130. La escritura del contenido actualizado en la estructura terciaria de datos 130 tiene lugar sin clasificar, por lo que se maximiza la velocidad de escritura. Todavía más tarde, cuando se van a hacer consultas en la base de datos, se lee el contenido actualizado del almacenamiento terciario 130, se clasifica y se escribe como contenido clasificado en la estructura secundaria de datos 120. Como se muestra en la figura 1, las estructuras terciaria y secundaria de datos 130, 120 son generalmente más pequeñas que la estructura primaria de datos 110. Líneas de trazos en las estructuras secundaria y terciaria de datos 120, 130, indican que la asignación de almacenamiento es dinámica en lugar de fija, y las estructuras de datos 120, 130 pueden crecer y contraerse según sea necesario.

La separación de la operación de clasificación de las operaciones de actualización, que maximiza la velocidad de escribir actualizaciones a costa de la velocidad de lectura del contenido actualizado, puede parecer antinatural en conexión con bases de datos OLAP. Pero la velocidad de lectura es más crucial en operaciones que implican grandes intervalos de datos, y dichas operaciones se pueden optimizar realizando la clasificación por separado de la actualización.

La figura 2 muestra esquemáticamente un diagrama de bloques de un sistema servidor de base de datos SS. Los dos bloques funcionales mayores del sistema servidor de base de datos SS son un ordenador servidor 200 de base de datos y un sistema de disco 290. El ordenador servidor 200 comprende una o más unidades de procesamiento central CP1,... CPn, generalmente indicadas por el numeral de referencia 210. Realizaciones que comprenden múltiples unidades de procesamiento 210 preferiblemente están provistas de una unidad de equilibrio de carga 215 que equilibra el procesamiento de carga entre las múltiples unidades de procesamiento 210. Las múltiples unidades de procesamiento 210 se pueden implementar como componentes separados de procesador o como núcleos físicos de procesador o procesadores virtuales dentro de una única carcasa de componentes. El ordenador servidor 200 comprende además una interfaz de red 220 para comunicarse con diversas redes de datos, que generalmente se indican por el signo de referencia DN. Las redes de datos DN puede incluir redes de área local, tal como una red Ethernet, y/o redes de área extensa, tal como internet. El sistema servidor SS da servicio a uno o más terminales cliente, indicados generalmente por el signo de referencia CT, por medio de las redes de datos DN.

El ordenador servidor 200 de la presente realización también comprende una interfaz de usuario 225. Dependiendo de la implementación, la interfaz de usuario 225 puede comprender circuitos locales de entrada-salida para una interfaz local de usuario, tales como un teclado, ratón y pantalla (no se muestran). Como alternativa o adicionalmente, la gestión del ordenador servidor 200 se puede implementar a distancia, utilizando la interfaz de red 220 y un terminal similar a los terminales cliente CT. La naturaleza de la interfaz de usuario depende de qué tipo de ordenador se use para implementar el ordenador servidor 200. Si el ordenador servidor 200 es un ordenador dedicado, puede no ser necesaria una interfaz local de usuario, y el ordenador servidor 200 puede ser gestionado a distancia, tal como desde un navegador web por internet, por ejemplo. Dicha gestión remota se puede conseguir por medio de la misma interfaz de red 220 que el ordenador servidor utiliza para tráfico entre él mismo y los terminales cliente.

El ordenador servidor 200 también comprende memoria 250 para almacenar instrucciones de programa, parámetros de funcionamiento y variables. El numeral de referencia 260 indica un paquete de programas para el ordenador servidor 200.

El ordenador servidor 200 también comprende circuitos para diversos relojes interruptores y similares, y estos se representan generalmente por el numeral de referencia 230. El ordenador servidor 200 comprende además una interfaz de disco 235 con el sistema de disco 290. Los diversos elementos 210 a 250 se intercomunican por medio de un bus 205, que lleva señales de dirección, señales de datos y señales de control, como conocen bien los expertos en la técnica.

El método inventivo se puede implementar en el sistema servidor SS de la siguiente manera. El paquete de programas 260 comprende instrucciones de código de programa para ordenar al conjunto de procesadores 210 para que ejecute las funciones del método inventivo, en donde las funciones incluyen configurar la estructura de datos 100 (consulte la figura 1) en el sistema de disco 290. Específicamente, las funciones del método inventivo incluyen las acciones definidas en la reivindicación 1.

La figura 3 explica un principio de funcionamiento de la invención. La siguiente descripción está relacionada con un caso en donde cada una de las estructuras primaria, secundaria y terciaria de datos se mantienen para cada columna. En algunas realizaciones las tablas de base de datos se pueden particionar horizontalmente, de manera que un intervalo de filas corresponde a una partición. En dichas realizaciones es beneficioso mantener las estructuras primaria, secundaria y terciaria de datos por separado para cada partición de una columna en cada tabla particionada. Implementar la invención en una tabla particionada no implica cambios en los dibujos, aparte del hecho de que las estructuras primaria, secundaria y terciaria de datos no corresponden a columnas enteras sino a particiones de las columnas.

El estado de las estructuras primaria, secundaria y terciaria de datos 110, 120, 130 se mostrará en cuatro fases de procesamiento distintas, designadas por las legras A, B, C y D, una de las cuales se adjuntará al numeral de referencia de la estructura de datos respectiva. Por ejemplo, el numeral de referencia 120C se refiere a la estructura secundaria de datos 120 en la fase de procesamiento C. Se debe entender que si bien datos en la estructura primaria de datos 110 se almacenan realmente en un formato comprimido, la presente explicación está relacionada con el contenido no comprimido, para permitir el seguimiento visual del flujo de información.

En la primera fase de procesamiento A, la estructura primaria de datos 110A se muestra con cinco elementos de datos de longitud variable numerados con las claves 1 a 5. Como se muestra, el contenido de los elementos de datos no tiene sentido, pero siguen un patrón que facilita el seguimiento del proceso a través de las fases A a D. En ese patrón, contenido original se muestra en letras mayúsculas, mientras que contenido actualizado se muestra en letras minúsculas. En la primera fase de procesamiento A, las estructuras secundaria y terciaria de datos 120A, 130A están vacías.

En la segunda fase de procesamiento B, el servidor de base de datos recibe tres sentencias de actualización y escribe los valores actualizados en la estructura terciaria de datos 130B. En el ejemplo ilustrado, se escribe contenido actualizado para las claves de almacenamiento 3, 1 y 4 (en ese orden) en la estructura terciaria de datos 130B. En esta fase, la estructura secundaria de datos 120B permanece vacía.

En la tercera fase de procesamiento C, se inicia una operación para leer los datos, y el motor de base de datos hace preparativos para ello. En la segunda fase de procesamiento B, la estructura terciaria de datos 130B está sin clasificar, por lo que leer su contenido requiere una búsqueda exhaustiva a través de la estructura terciaria de datos 130. Con el fin de acelerar las operaciones de lectura, se lee el contenido actualizado para claves de almacenamiento 3, 1 y 4 de la estructura terciaria de datos 130B, se clasifica por la clave de almacenamiento y se escribe en la estructura secundaria de datos 120C. La estructura terciaria de datos 130C se queda vacía. La acción de leer contenido de la estructura terciaria de datos, clasificar y escribir en la estructura secundaria de datos son acciones preparativas para leer los datos en la columna particular de la tabla particular. Finalmente, cuando la base de datos se lee realmente, el proceso de lectura se ilustra por el numeral de referencia 140C. En la etapa de lectura representada por el numeral de referencia 140C, primero se leen elementos de datos para las claves 1 a 5 de la estructura primaria de datos 110C, y si una versión actualizada para los elementos de datos está almacenada en la estructura secundaria de datos 120C, esa versión actualizada de la estructura secundaria de datos 120C sobrescribe a la versión original de la estructura primaria de datos 110C. Un resultado equivalente se obtiene de una operación alternativa de lectura en donde primero se buscan elementos de datos en la estructura secundaria de datos 120C, y si la estructura secundaria de datos 120C no tiene una entrada para un elemento de datos (tales como los elementos 2 y 5 en el presente ejemplo), los elementos de datos correspondientes se leen de la estructura primaria de datos 110C. El numeral de referencia 140C representa el resultado de la etapa de lectura para las claves 1 a 5, en donde los elementos de datos con claves 1, 3 y 4 (letras minúsculas) se leen de la estructura secundaria de datos 120C, mientras que los elementos de datos con claves 2 y 5 (letras mayúsculas) se leen de la estructura primaria de datos 110C.

La cuarta fase de procesamiento D ilustra una operación de confirmación de transacción. En esta operación, se combinan los datos de las estructuras primaria y secundaria de datos, dando como resultado una versión nueva de los datos primarios comprimidos, indicados por el numeral de referencia 110D. En la operación de combinación, los valores para claves de almacenamiento 1, 3 y 4 se leen de la estructura secundaria de datos 120C, mientras que los valores para claves de almacenamiento 2 y 5 se leen de la estructura primaria de datos 110C. Los datos combinados se escribirán en un formato comprimido en la versión nueva de la estructura primaria de datos comprimidos 110D. Las estructuras secundaria y terciaria de datos 120D y 130D se quedan vacías.

La figura 4 ilustra aislamiento mutuo de transacciones concurrentes, es decir, transacciones que se superponen al menos parcialmente en el tiempo. El escenario mostrado en la figura 3 se representa para una transacción

únicamente, y las cosas parecen bastante diferentes para una transacción diferente, concurrente. Esto se ilustra en la figura 4.

5 El numeral de referencia 110' representa de nuevo la estructura primaria de datos. Pero las estructuras secundaria y terciaria de datos se implementan por separado para cada transacción, dos de la cuales se muestran en la figura 4. (Es obvio que dos transacciones son suficientes para ilustrar el aislamiento de transacciones, pero de otro modo el número es puramente arbitrario). Los numerales de referencia 420<sub>1</sub> y 430<sub>1</sub> representan, respectivamente, las estructuras secundaria y terciaria de datos para una primera transacción, mientras que las estructuras de datos correspondientes para la segunda transacción se representan por los numerales de referencia 420<sub>2</sub> y 430<sub>2</sub>. Cuando la primera transacción lee los elementos de datos 1 a 5, los resultados se representan por el numeral de referencia 440<sub>1</sub>. De manera similar, cuando la segunda transacción lee los elementos de datos 1 a 5, los resultados se representan por el numeral de referencia 440<sub>2</sub>. Como se muestra en la figura 4, los elementos de datos sin cambiar por una transacción se leen de la estructura primaria de datos 110', mientras que los elementos de datos cambiados por una transacción se leen de la estructura secundaria de datos de la transacción en cuestión. Así todas las transacciones están aisladas entre sí. (Los expertos entenderán que en el presente ejemplo, la relación de datos cambiados a datos no cambiados es puramente arbitraria).

15 Para un experto en la técnica será evidente que, a medida que avanza la tecnología, el concepto inventivo puede ser implementado de diversas maneras. La invención y sus realizaciones no se limitan a los ejemplos descritos anteriormente, sino que pueden variar dentro del alcance de las reivindicaciones.

#### Referencias

- 20 1. "C-Store: A Column Oriented DBMS", Mike Stonebraker, Daniel Abadi, Adam Batkin, Xuedong Chen, Mitch Cherniack, Miguel Ferreira, Edmond Lau, Amerson Lin, Sam Madden, Elizabeth O'Neil, Pat O'Neil, Alex Rasin, Nga Tran and Stan Zdonik. VLDB, páginas 553-564, 2005.

**REIVINDICACIONES**

1. Un método para manejar una base de datos electrónica, el método comprende las siguientes etapas realizadas por un ordenador servidor (200) para cada una de una o más columnas en cada una de una o más tablas:

en una fase de preparación de base de datos:

- 5           - asignar al menos una estructura primaria de datos (110), al menos una estructura secundaria de datos (120), y al menos una estructura terciaria de datos (130);

en una fase de escritura de base de datos:

- escribir contenido en columnas en un formato comprimido en la al menos una estructura primaria de datos (110);

10          en una fase de actualización de base de datos:

- escribir un valor actualizado para cada columna de cada fila actualizada en la al menos una estructura terciaria de datos (130) de la columna respectiva, junto con una clave de almacenamiento que indica una fila correspondiente en la al menos una estructura primaria de datos (110);

en una fase de consulta de base de datos:

- 15           - leer los valores actualizados y las claves de almacenamiento de la al menos una estructura terciaria de datos (130), clasificar los valores actualizados por sus claves de almacenamiento y almacenar los valores clasificados y las claves de almacenamiento en el orden clasificado en la al menos una estructura secundaria de datos (120), y limpiar la al menos una estructura terciaria de datos (130);

- 20           - responder a una consulta de base de datos preguntando a la al menos una estructura primaria de datos (110) y a la al menos una estructura secundaria de datos (120);

en una fase de confirmación de base de datos:

- producir una versión nueva (110D) de la al menos una estructura primaria de datos (110) combinando la al menos una estructura secundaria de datos (120) y una versión antigua (110C) de la al menos una estructura primaria de datos;

25          en donde la pregunta y la combinación comprenden leer un valor para una clave de almacenamiento de la al menos una estructura secundaria de datos (120) si el valor existe o de lo contrario de la al menos una estructura primaria de datos (110).

30          2. El método según la reivindicación 1, que comprende además: mantener la estructura secundaria de datos (420<sub>1</sub>, 420<sub>2</sub>) y la estructura terciaria de datos (430<sub>1</sub>, 430<sub>2</sub>) por separado para cada una de varias transacciones concurrentes.

3. El método según la reivindicación 1, que comprende además: mantener, para una tabla particionada horizontalmente, una estructura primaria de datos, una estructura secundaria de datos y una estructura terciaria de datos separadas para cada partición horizontal de cada una de varias columnas de la tabla particionada horizontalmente.

35          4. Un ordenador servidor (200) de base de datos para manejar una base de datos electrónica (290), el ordenador servidor de base de datos comprende

- al menos una unidad de procesamiento (CP1,... CPn)

- memoria (250) para almacenar aplicaciones y datos;

40           - en donde la memoria (250) comprende instrucciones de código de programa para ordenar a la al menos una unidad de procesamiento que lleve a cabo las siguientes etapas para cada de una o más columnas en cada una de una o más tablas:

en una fase de preparación de base de datos:

- asignar al menos una estructura primaria de datos (110), al menos una estructura secundaria de datos (120), y al menos una estructura terciaria de datos (130);

45          en una fase de escritura de base de datos:

- escribir contenido en columnas en un formato comprimido en la al menos una estructura primaria de datos (110);

en una fase de actualización de base de datos:

- 5
- escribir un valor actualizado para cada columna de cada fila actualizada en la al menos una estructura terciaria de datos (130) de la columna respectiva, junto con una clave de almacenamiento que indica una fila correspondiente en la al menos una estructura primaria de datos (110);

en una fase de consulta de base de datos:

- 10
- leer los valores actualizados y las claves de almacenamiento de la al menos una estructura terciaria de datos (130), clasificar los valores actualizados por sus claves de almacenamiento y almacenar los valores clasificados y las claves de almacenamiento en el orden clasificado en la al menos una estructura secundaria de datos (120), y limpiar la al menos una estructura terciaria de datos (130);

responder a una consulta de base de datos preguntando a la al menos una estructura primaria de datos (110) y a la al menos una estructura secundaria de datos (120);

en una fase de confirmación de base de datos:

- 15
- producir una versión nueva (110D) de la al menos una estructura primaria de datos (110) combinando la al menos una estructura secundaria de datos (120) y una versión antigua (110C) de la al menos una estructura primaria de datos;

en donde la pregunta y la combinación comprenden leer un valor para una clave de almacenamiento de la al menos una estructura secundaria de datos (120) si el valor existe o de lo contrario de la al menos una estructura primaria de datos (110).

25

5. El ordenador servidor de base de datos según la reivindicación 4, en donde la memoria (250) comprende además instrucciones de código de programa para ordenar a la al menos una unidad de procesamiento que mantenga la estructura secundaria de datos (4201, 4202) y la estructura terciaria de datos (4301, 4302) por separado para cada una de varias transacciones concurrentes.

25

6. El ordenador servidor de base de datos según la reivindicación 4, o 5, en donde la memoria (250) comprende además instrucciones de código de programa para ordenar a la al menos una unidad de procesamiento que mantenga, para una tabla particionada horizontalmente, una estructura primaria de datos, una estructura secundaria de datos y una estructura terciaria de datos separadas para cada partición horizontal de cada una de varias columnas de la tabla particionada horizontalmente.

30

7. El ordenador servidor de base de datos según las reivindicaciones 4, 5 o 6, en donde el ordenador servidor de base de datos comprende múltiples unidades de procesamiento (CP1,... CPn) y una unidad de equilibrio de carga (215) para distribuir la carga de procesamiento entre las múltiples unidades de procesamiento.

35

8. Un soporte tangible de programa informático, que tiene un conjunto de instrucciones de programa informático codificado sobre el mismo, que son ejecutables en un ordenador servidor (200) de base de datos para manejar una base de datos electrónica (290), en donde el ordenador servidor de base de datos comprende al menos una unidad de procesamiento (CP1,... CPn) y una memoria (250) para almacenar aplicaciones y datos;

en donde la ejecución de las instrucciones de programa informático en el ordenador servidor de base de datos provoca que el ordenador servidor de base de datos lleve a cabo las siguientes etapas para cada una de una o más columnas en cada una de una o más tablas: en una fase de preparación de base de datos:

40

asignar al menos una estructura primaria de datos (110), al menos una estructura secundaria de datos (120), y al menos una estructura terciaria de datos (130);

en una fase de escritura de base de datos:

- escribir contenido en columnas en un formato comprimido en la al menos una estructura primaria de datos (110);

45

en una fase de actualización de base de datos:

- escribir un valor actualizado para cada columna de cada fila actualizada en la al menos una estructura terciaria de datos (130) de la columna respectiva, junto con una clave de almacenamiento que indica una fila correspondiente en la al menos una estructura primaria de datos (110);

en una fase de consulta de base de datos:

- 5 - leer los valores actualizados y las claves de almacenamiento de la al menos una estructura terciaria de datos (130), clasificar los valores actualizados por sus claves de almacenamiento y almacenar los valores clasificados y las claves de almacenamiento en el orden clasificado en la al menos una estructura secundaria de datos (120), y limpiar la al menos una estructura terciaria de datos (130);
- responder a una consulta de base de datos preguntando a la al menos una estructura primaria de datos (110) y a la al menos una estructura secundaria de datos (120);

en una fase de confirmación de base de datos:

- 10 - producir una versión nueva (110D) de la al menos una estructura primaria de datos (110) combinando la al menos una estructura secundaria de datos (120) y una versión antigua (110C) de la al menos una estructura primaria de datos;

en donde la pregunta y la combinación comprenden leer un valor para una clave de almacenamiento de la al menos una estructura secundaria de datos (120) si el valor existe o de lo contrario de la al menos una estructura primaria de datos (110).

15 9. El soporte tangible de programa informático según la reivindicación 8, en donde la ejecución de las instrucciones de programa informático en el ordenador servidor de base de datos provoca que el ordenador servidor de base de datos mantenga la estructura secundaria de datos (420<sub>1</sub>, 420<sub>2</sub>) y la estructura terciaria de datos (430<sub>1</sub>, 430<sub>2</sub>) por separado para cada de varias transacciones concurrentes.

20 10. El soporte tangible de programa informático según la reivindicación 8, en donde la ejecución de las instrucciones de programa informático en el ordenador servidor de base de datos provoca que el ordenador servidor de base de datos mantenga, para una tabla particionada horizontalmente, una estructura primaria de datos, una estructura secundaria de datos y una estructura terciaria de datos separadas para cada partición horizontal de cada una de varias columnas de la tabla particionada horizontalmente.

Fig. 1

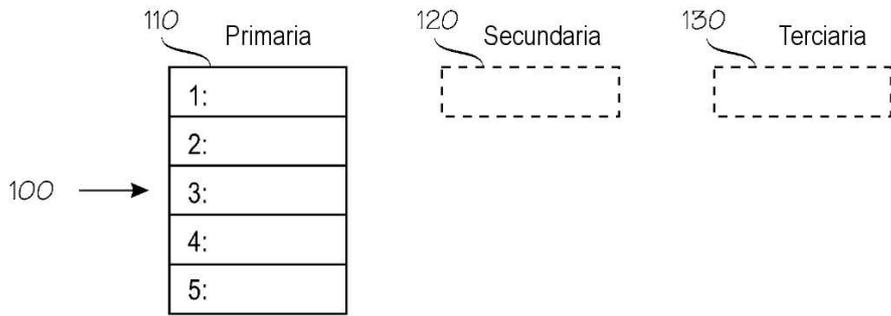


Fig. 2

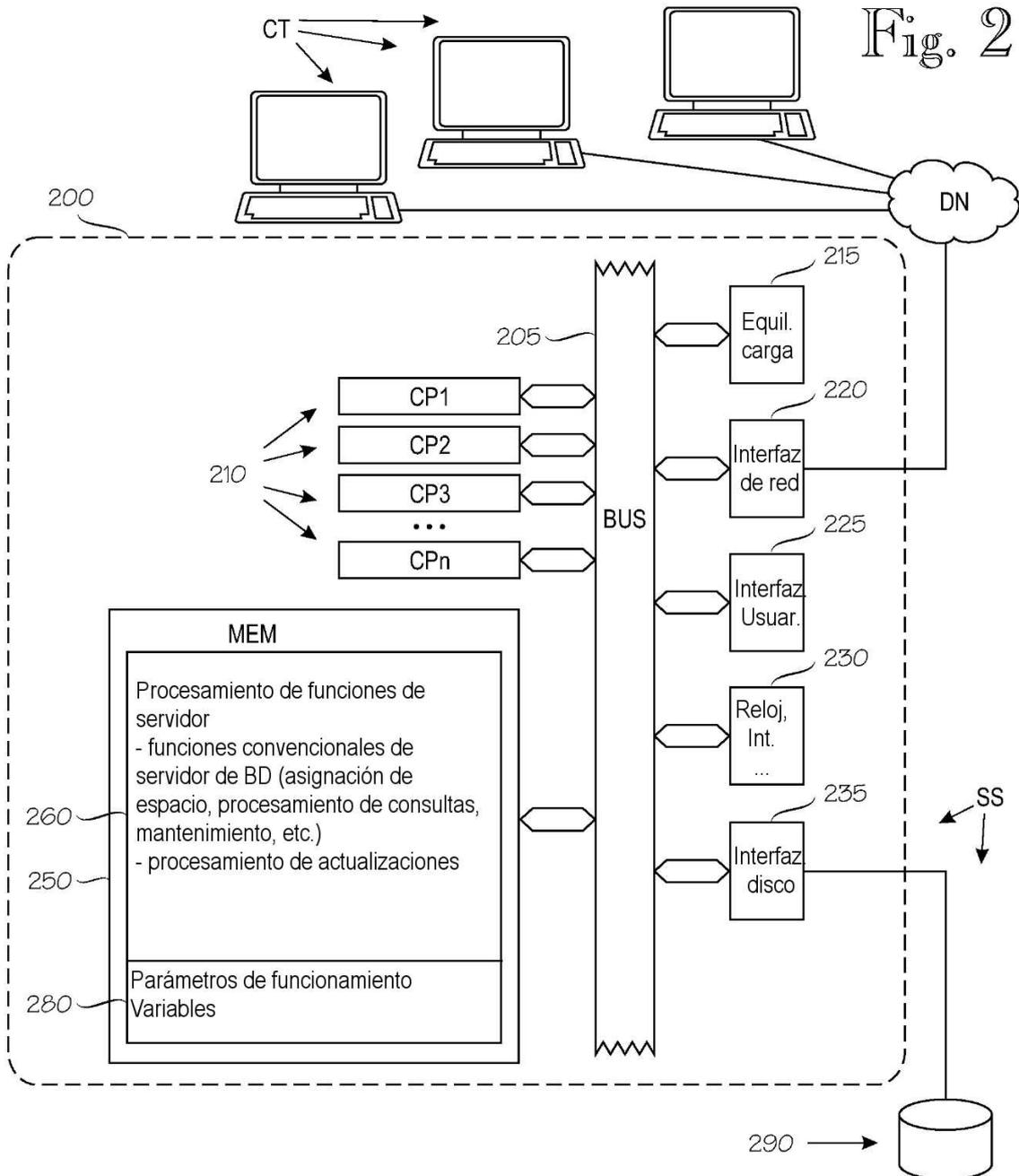


Fig. 3

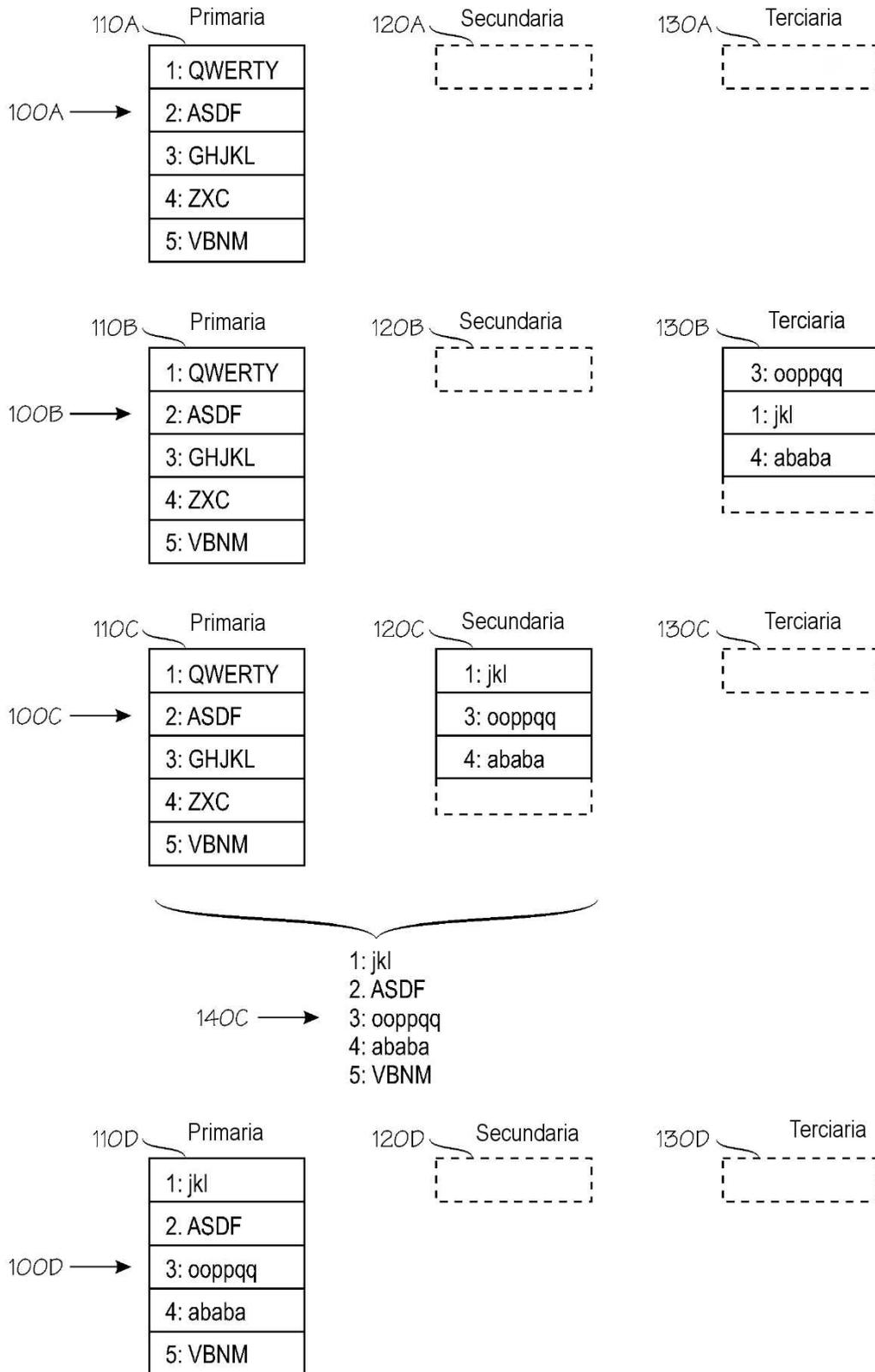


Fig. 4

