

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 668 723**

21 Número de solicitud: 201631481

51 Int. Cl.:

G06F 17/30 (2006.01)

12

SOLICITUD DE PATENTE

A1

22 Fecha de presentación:

18.11.2016

43 Fecha de publicación de la solicitud:

21.05.2018

71 Solicitantes:

8KDATA TECHNOLOGY, S. L. (100.0%)
Carretera de Fuencarral, 44 Edificio 4B, Loft 33
28108 ALCOBENDAS (Madrid) ES

72 Inventor/es:

HERNÁNDEZ TORTOSA, Álvaro;
DARIAS CAMACHO, Yeray;
MELLI, Matteo;
ORTIZ JAUREGUIZAR, Gonzalo y
LÓPEZ BEZANILLA, Jerónimo

74 Agente/Representante:

SALVA FERRER, Joan

54 Título: **MÁQUINA DE EXTRACCIÓN, MAPEO Y ALMACENAMIENTO DE DATOS JERÁRQUICOS**

57 Resumen:

La presente divulgación referencia sistemas y métodos y productos de programas computacionales para procesar datos de documentos jerárquicos (por ejemplo, documentos XML y JSON) y almacenarlos en estructuras de bases de datos relacionales. Los métodos presentados en este documento leen los datos de los documentos de entrada y analizan el esquema del documento para crear tanto las tablas dinámicas de datos así como las tablas de metadatos para el almacenamiento de la estructura tabular de destino. Después de esto, los datos de clave-valor almacenados en el documento son extraídos, transformados, y mapeados a una estructura de tablas generada de las cuales se hace referencia en las tablas de metadatos. De este modo, la información puede ser almacenada en bases de datos o en estructuras tabulares o relacionales y reconstruida en el documento original si fuese necesario.

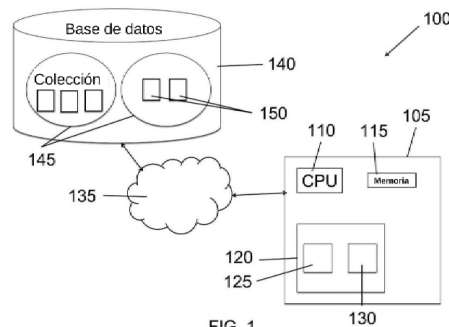


FIG. 1

DESCRIPCIÓN

MÁQUINA DE EXTRACCIÓN, MAPEO Y ALMACENAMIENTO DE DATOS JERÁRQUICOS

5

La invención aquí descrita está relacionada en general con sistemas y métodos para almacenar, transformar y extraer datos, desde o hacia unas fuentes de datos, que utilizan, al menos en parte, estructuras tabulares. En particular, la invención está relacionada con sistemas y métodos para almacenar, transformar y extraer datos desde documentos almacenados en fuentes de datos
10 estructuradas y no estructuradas.

Antecedentes de la invención

Las bases de datos NoSQL, los sistemas de procesado y almacenamiento de datos
15 semiestructurados, y otro software llamado schema-less (sin esquema), aceptan estructuras de datos de clave-valor, o “documentos”, como entrada. Estos documentos son una forma conveniente de representar los datos de naturaleza jerárquica, en la cual hay pocas o ninguna restricción sobre el contenido o la estructura (esquema) de los datos. En particular, la estructura, los contenidos y las claves de distintos documentos, aunque estén lógicamente agrupados,
20 pueden ser completamente distintos entre ellos.

La estructura de un documento puede incluir un conjunto de pares clave-valor, en el cual la clave es un nombre (por ejemplo, una cadena de texto) y el valor puede ser un escalar (por ejemplo, números, texto, booleans, valores vacíos, etc.) o un valor compuesto. Los valores compuestos
25 incluyen valores anidados, como puede ser un documento incrustado o una colección de otros escalares o valores compuestos. Para algunos procesos software en particular, los pares clave-valor pueden tener forma de una única e indivisible unidad almacenada de estructuras clave-valor anidadas. Los datos (por ejemplo, el valor) contenidos en el documento se acceden típicamente a través de una o más claves (aunque una clave no es necesariamente requerida para acceder
30 a los datos), que pueden ser claves generadas o claves extraídas y/o copiadas desde el documento original y que identifican el documento unívocamente.

El acceso convencional a los datos de los documentos se efectúa accediendo (por ejemplo, consultando) a los datos a través de un índice (externo) que indexa la clave primaria u otro campo o campos del documento. De todos modos, el acceso convencional a datos de los documentos no es siempre deseable, óptimo, o posible en ciertos escenarios de acceso a los datos, como en
5 consultas no indexadas, consultas agregadas o datos en campos anidados o documentos anidados dentro de los datos. En estas situaciones, los métodos implementados para el acceso a los datos de los documentos utilizan los recursos de las computadoras de forma ineficiente (por ejemplo, largos tiempos de CPU y alto uso de la memoria). Para ser más explícitos, una consulta no indexada a una colección de documentos requiere una operación de lectura total de la
10 colección. Las operaciones de lectura total de la colección requieren el análisis de todos los datos documento por documento, clave por clave, aplicando el predicado de la consulta por cada documento. Esta operación incluye los inconvenientes técnicos del uso de largos tiempos de procesado, la creación de frecuentes cuellos de botella en el I/O la CPU, y expone una mala pauta de utilización de la cache.

15 Además de lo anteriormente mencionado, los actuales sistemas encuentran dificultades cuando procesan documentos que tienen un esquema poco eficiente o son schema-less (sin esquema), debido al hecho de que la estructura utilizada debe ser definida para cada documento. Si los documentos dentro de un conjunto dado comparten una estructura subyacente igual o
20 suficientemente similar, se incurre en un significativo sobreesfuerzo computacional por la innecesaria repetición del esquema, que lleva a un incorrecto uso de espacio, memoria y procesamiento.

Por eso, existe una necesidad de sistemas y métodos para extraer, transformar y almacenar los
25 datos de documentos desde una fuente de datos en la cual los documentos tienen un esquema variable o son schema-less (sin esquema), para mejorar la gestión de los recursos computacionales. Además, existe la necesidad de sistemas y métodos que puedan procesar eficientemente datos jerárquicos de clave-valor anidados.

30 Es respecto a estos problemas y otros que se proporciona la presente invención.

Descripción de la invención

Por una parte, las realizaciones de la invención se refieren al método para mapear uno o más pares clave-valor, asociados con un documento, en una o más estructuras tabulares, los pares clave-valor tiene un nombre de clave y un valor, y cada una de dichas una o más estructuras tabulares tienen una o más filas y columnas para almacenar valores. Por ejemplo, la estructura tabular puede ser un almacenamiento persistente. De acuerdo con una o más realizaciones, el método comprende leer el documento, con un lector de documentos, para identificar los pares clave-valor asociados con el documento de entrada. Por ejemplo, el documento puede entrar desde una fuente de datos, como una base de datos relacional, y puede ser de muchos formatos, como JSON o XML. Luego, el método determina si un valor asociado con un dato para los pares clave-valor es un valor escalar o un valor compuesto.

En el caso en que el valor asociado con el par clave-valor sea un valor escalar, el método efectúa los pasos de extraer el nombre de la clave del par clave-valor y almacena el valor del par clave-valor en una fila de la estructura tabular. En una realización, el método además efectúa el paso de comprobar cuando la estructura tabular tiene una columna asociada con el nombre de la clave extraída desde el par clave-valor. Esto puede incluir generar una nueva columna asociada con la estructura tabular que almacene el valor del par clave-valor en una fila de la estructura tabular. En algunas realizaciones, la columna es identificada por un tipo asociado con el valor del par clave-valor.

En el caso de que el valor asociado con el par clave-valor es un valor compuesto, el método efectúa los pasos de extraer el nombre de la clave del par clave-valor y generar una subestructura tabular asociada con el nombre de la clave del par clave-valor. En algunas realizaciones, en el caso de que un valor asociado con el par clave-valor sea un valor compuesto, el método además efectúa el paso de enviar el valor del par clave-valor a una estructura de datos temporal. En otras realizaciones, el método efectúa el paso de iterar recursivamente los precedentes hasta que cada uno de los pares clave-valor hayan sido extraídos a una estructura tabular o subestructura tabular.

Además, el método en una o más realizaciones adicionalmente incluye el procesado de los pares clave-valor contenidos en valores compuestos que son enviados a estructuras de datos temporales. Por ejemplo, la estructura de datos temporal puede ser una pila de operaciones o una lista enlazada. En particular, el método determina cuando el valor compuesto asociado con el par clave-valor enviado a la estructura de datos temporal incluye uno o más subpares clave-

valor. Además, el método determina cuando el o los subpares clave-valor son un valor escalar o un valor compuesto para cada uno de los subpares clave-valor. En el caso de que el valor asociado con el subpar clave-valor sea determinado como valor escalar, el método efectúa los pasos de extraer el nombre de la clave del subpar clave-valor y comprobar si la subestructura tabular tiene asociada una columna con el nombre del subpar clave-valor.

Continuando con lo anterior, si la comprobación es correcta, el método continúa efectuando el paso de almacenar el valor del subpar clave-valor en una fila de la subestructura tabular y mapea el valor del subpar clave-valor a la columna. Si no, el método efectúa el paso de generar una nueva columna asociada con la estructura subtabular, almacenando el valor del subpar clave-valor en una fila de la subestructura tabular y mapeando el valor del subpar clave-valor a la nueva columna. En el caso de que el valor asociado con el subpar clave-valor sea un valor compuesto, el método efectúa los pasos de extraer el nombre de la clave del subpar clave-valor, y envía el valor del subpar clave-valor a la estructura de datos temporal. En el caso de que el valor asociado con el subpar clave-valor no sea un valor compuesto, el método efectúa el paso de eliminar el par clave-valor enviado desde la estructura de datos temporal. Los pasos anteriores pueden ser iterados hasta que la estructura de datos temporal no contenga ningún otro par clave-valor enviado anteriormente.

Visto de otra forma, las realizaciones de la invención se refieren a sistemas para extraer y almacenar los datos de documentos en una base de datos. En una o más realizaciones, el sistema incluye un aparato de análisis de datos que incluye un procesador y una memoria acoplada al procesador. Además, el sistema incluye una fuente de datos que contiene uno o más documentos. Cada documento contiene uno o más pares clave-valor, y cada par clave-valor tiene un nombre de clave y un valor, en el cual cada valor tiene un tipo de valor. Además, el sistema incluye un lector de documentos para recibir los documentos desde la fuente de datos a través de una red, el lector de documentos estará acoplado al aparato de análisis de datos. Adicionalmente, el sistema incluye un módulo de extracción de estructura.

El módulo de extracción de estructura implementa un código de programa del procesador para generar una o más estructuras tabulares, que tienen al menos una columna correspondiente con el nombre de la clave y el tipo de valor de cada uno de los pares clave-valor. Así mismo, el sistema incluye un módulo de extracción de datos. El módulo de extracción de estructura implementa un

código de programa del procesador para extraer el valor en una fila de datos en las estructuras tabulares creadas por el mismo módulo de extracción de datos. Finalmente, el sistema incluye una o más tablas de metadatos generadas por el módulo de extracción de datos con referencias a una o más estructuras tabulares.

5

Breve descripción de los diseños

La invención está ilustrada en las figuras de los diseños adjuntos, que tienen el objetivo de ser unos ejemplos y no unas limitaciones, en los cuales las referencias tienen el objetivo de ser referencias a las partes correspondientes o similares, y en las cuales:

10

La Fig. 1 presenta un diagrama de bloques que ilustra un sistema para mapear datos de documentos a una estructura tabular de acuerdo con una realización de la invención presentada;

15 La Fig. 2 presenta una ilustración que muestra una jerarquía de pares clave-valor de ejemplo de acuerdo con una realización de la presente invención;

La Fig. 3 presenta un flujo de ejemplo del método para mapear datos de documentos a una estructura tabular de acuerdo con una realización de la presente invención;

20

La Fig. 4 presenta un flujo del método de mapeo de uno o más pares clave-valor asociado con un documento en una o más estructuras tabulares de acuerdo con una realización de la presente invención;

25 La Fig. 5 presenta un flujo del método para generar las columnas de las estructuras tabulares de acuerdo con una realización de la presente invención;

La Fig. 6 presenta una ilustración que muestre la resolución de los conflictos de tipo de los valores de acuerdo con una realización de la presente invención;

30

La Fig. 7 presenta una ilustración que muestra la generación de las tablas de los metadatos de acuerdo con una realización de la presente invención;

La Fig. 8 presenta una estructura de documento de ejemplo que contiene múltiples pares clave-valor para ser procesados en una estructura tabular de acuerdo con una realización de la presente invención;

5 La Fig. 9 presenta una ilustración que muestra los datos del documento de la Fig. 8 mapeados a una estructura tabular generada de acuerdo con una realización de la presente invención;

La Fig. 10 presenta una ilustración que muestra el procesado de la raíz de un documento de ejemplo de la Fig. 8 de acuerdo con una realización de la presente invención;

10

La Fig. 11 presenta una ilustración que muestra el procesado del primer subnivel del documento de ejemplo de la Fig. 8 de acuerdo con una realización de la presente invención;

La Fig. 12 presenta la ilustración que muestra el procesado del segundo subnivel del documento de ejemplo de la Fig. 8 de acuerdo con una realización de la presente invención; y

15

La Fig. 13 presenta la ilustración que muestra una operación de ejemplo sobre la pila de acuerdo con una realización de la presente invención;

20 **Descripción de una realización preferida**

A lo largo de la memoria y de las reivindicaciones, los términos pueden tener un determinado matiz, sugerido o implícito en el contexto más allá de un significado escrito explícitamente. Asimismo, la frase “en una implementación” o “en una realización” utilizadas en este documento no se refieren necesariamente a una implementación o realización diferente. De modo similar, la frase “una o más implementaciones” o “una o más realizaciones” utilizadas en este documento no se refieren necesariamente a la misma implementación o realización y la frase “al menos una implementación” o “al menos una realización” utilizadas en este documento no se refieren a una implementación o realización diferente. La intención es, por ejemplo, que las reivindicaciones que son materia del objeto de este documento incluyan las combinaciones de ejemplo de las implementaciones y realizaciones en su todo o en parte.

30

En el presente documento se proporcionan unos sistemas y unos métodos para mapear datos asociados con un documento, por ejemplo uno o más pares clave-valor, en una o más estructuras tabulares u otros formatos de datos relacionales. En este documento, un “documento” es un conjunto, no necesariamente ordenado, de pares clave-valor, donde la clave es un identificador de datos (por ejemplo, un nombre) y el valor es o un valor escalar o un valor compuesto. Los valores compuestos pueden ser documentos anidados heterogéneos u homogéneos o colecciones de otros valores escalares y/o compuestos. Por ejemplo, un valor compuesto puede ser una agrupación ordenada o un subdocumento contenido dentro de un documento. Los documentos incluyen datos almacenados en uno o más niveles del documento. Por ejemplo, cada documento tiene un nivel raíz o base. Después de eso, el documento incluye al menos un subnivel para cada par clave-valor almacenado en el nivel raíz y que tenga un valor compuesto. Los subniveles pueden incluir posteriores pares clave-valor compuestos y por eso el documento puede incluir tantos niveles como subdocumentos anidados tenga. Además, los documentos pueden ser agrupados en una o más colecciones que, a su vez, pueden ser agrupados en bases de datos. Una colección puede incluir documentos agrupados que tienen una relación entre ellos (por ejemplo, el mismo tipo de documento) o documentos agrupados que no tienen ninguna relación entre ellos.

En particular, los sistemas y métodos descritos en este documento mejoran las prácticas convencionales para almacenar datos de documentos no estructurados transformando datos de documentos no estructurados en datos estructurados tabulares u otros sistemas de datos relacionales. Una estructura tabular es un posible almacenamiento persistente que agrupa los datos en tablas. Una estructura tabular está compuesta de filas y columnas. Cada registro o tupla es almacenado en una fila, y la columna define cada atributo de la fila con un nombre y un tipo. Por ejemplo, el valor de un par clave-valor es almacenado en una fila de una tabla y está asociado con una columna en particular que se define de acuerdo con el nombre de la clave y el tipo del valor (por ejemplo, integer, double, string, character, boolean, etc. - donde integer es un tipo numérico entero, float es un tipo numérico representado con coma flotante y double lo mismo que el anterior pero de doble precisión, string es una cadena de texto, character es tipo de dato que representa un único carácter y boolean en un tipo de dato de lógica binaria que puede tener valor verdadero o falso) del par clave-valor.

En un aspecto de la presente invención, los sistemas y los métodos aquí mostrados y descritos incluyen dos procesamientos de datos paralelos. Primero, la presente aplicación detalla los métodos para extraer estructuras de datos desde un documento y después para generar dinámicamente y/o actualizar una estructura tabular (por ejemplo, tablas) para representar el contenido de los datos del documento basado en la estructura extraída de los datos. En una o más realizaciones, los métodos aquí proporcionados definen la información de los metadatos en las tablas de metadatos correspondientes, tal información de metadatos referencia las estructuras extraídas de los datos, de modo que la estructura tabular se actualiza dinámicamente, de acuerdo con los nuevos datos del documento de entrada conforme con la correcta referencia con los datos ya almacenados. Segundo, la presente aplicación detalla los métodos para leer los datos de los documentos, para crear y almacenar los registros de datos en las estructuras tabulares generadas por el primer proceso de procesado de datos. Esto incluye los documentos que tienen valores compuestos, como unos documentos anidados (por ejemplo, un par clave-valor que tiene un valor que es el también un documento). El presente método aquí mostrado y descrito genera tablas adicionales para cada par clave-valor que tiene documentos anidados.

La información se representa como un conjunto de datos tabulares o tablas relacionales en un sistema relacional de acuerdo con el diseño del esquema del sistema. Clasificar los datos de esta forma permite consultas sin efectuar una lectura completa de la base de datos, dado que únicamente una parte de las tablas o el subconjunto de los datos en las tablas son el objetivo de los criterios de la consulta, esto permite mejorar las prestaciones de la consulta, disminuir los cuellos de botella de I/O y de la CPU y mejorar la utilización de la cache, comparado con las practicas convencionales para manejar datos de documentos no estructurados jerárquicos o anidados. Por ejemplo, un sistema tabular o relacional, como el aquí mostrado y descrito, define la información de la tabla y la de la columna una única vez, lo cual permite alcanzar una reducción del almacenamiento y de la cache considerable comparado con los datos almacenados por las bases de datos NoSQL convencionales.

La Fig. 1 ilustra un sistema 100 para mapear los datos de una fuente de datos a una estructura tabular de acuerdo con una o más realizaciones de la presente invención. El sistema 100 incluye un aparato de procesado de los datos 105 que tiene un procesador 110, una memoria 115 acoplada con el procesador, y uno o más módulos software 120 que implementan, a través del procesador, un código de programa almacenado en la memoria para efectuar aspectos del mapeo

de datos como los aquí mostrados y descritos. Por ejemplo, los módulos de software 120 pueden incluir un extractor de estructura 125 para generar estructuras tabulares para almacenar los datos y un extractor de datos 130 para extraer los datos contenidos en un documento y almacenar esos datos extraídos en una estructura tabular generada. El aparato de procesado de datos 105 puede
5 incluir, por ejemplo, servidores, ordenadores portátiles y/o ordenadores de escritorio, y dispositivos de computación como dispositivos de computación en formato tabletas, teléfonos celulares inteligentes, asistentes digitales personales o similares. La memoria 115 puede ser utilizada para almacenar los datos, los metadatos, y los programas para ser ejecutados por el procesador 110. La memoria 115 puede incluir una o más memorias volátiles y no volátiles, como
10 una memoria de acceso aleatorio (Random Access Memory o “RAM”), memoria de solo lectura (Read Only Memory o “ROM”), memoria flash, memoria de cambio de fase (Phase Change Memory o “PCM”), u otros tipos.

El aparato que procesa los datos 105 está configurado para acceder a la fuente de datos 140. La
15 fuente de datos 140 puede ser local al aparato de procesado de datos 105, o remota, en cuyo caso los dos están conectado a través de una red 135 (por ejemplo, una red cableada o sin hilos, redes 3G/4G, etc.). En una o más implementaciones, la fuente de datos 140 es una base de datos. Por ejemplo, la fuente de datos 140 puede ser una base de datos relacional. Los datos almacenados en la fuente de datos 140 son almacenados en una o más colecciones 145, cada
20 colección tiene uno o más documentos 150. Los documentos 150 incluyen datos almacenados no estructurados a lo largo de una ruta en el documento en la forma de un conjunto de pares clave-valor, en el cual cada par clave-valor incluye una clave (o “nombre de clave”) que es un identificador que representa un valor del dato asociado con esa clave. El valor del dato (o “valor”) puede ser un valor escalar (por ejemplo, un integer, double, string, character, float, boolean, valor vacío, etc.) o un valor compuesto (por ejemplo, una agrupación ordenada, un registro, un
25 conjunto, una función, un valor anidado, etc.). En algunos casos, un par clave-valor puede ser un string, por ejemplo, el nombre de un libro, y un valor correspondiente a ese string, por ejemplo, “The Martian”. Un documento se estructura de forma que primero contiene los pares clave-valor en un nivel raíz. Si un documento incluye datos anidados en un nivel raíz (o cualquier subnivel),
30 o sea, un par clave-valor que tiene un valor compuesto (por ejemplo, otro par clave valor), entonces el dato es almacenado en un subnivel. En una o más realizaciones, un documento 150 es almacenado en un formato particular de intercambio de datos. Por ejemplo, el documento 150

puede ser almacenado en JAVASCRIPT Object Notation (“JSON”), Extensible Markup Language (“XML”), Resource Description Framework (“RDF”), YAML, Rebol, o Gellish.

La Fig. 2 ilustra la estructura de un par clave-valor jerárquico de un documento de ejemplo 200 de acuerdo a una o más realizaciones de este documento. En este documento de ejemplo 200, un nivel raíz 205 incluye cuatro o más pares clave-valor, denotados por K1, K2, K3 y K4. El par clave-valor K1 tiene un valor V1, que es un valor compuesto. En el ejemplo ilustrado, V1 contiene dos pares clave-valor anidados, K5 y K6, que son almacenados en un primer subnivel 210. El par clave-valor K2 tiene un valor de V2, que es un valor escalar, y en consecuencia no tiene subnivel asociado con él. El par clave-valor K3 tiene un valor vacío asociado, que también quiere decir que no tiene subnivel asociado con él. El par clave-valor K4 tiene un valor de [V7, V8], que es una agrupación ordenada (un valor compuesto). Las agrupaciones ordenadas son almacenadas en un subnivel, y la Fig. 2 ilustra que la agrupación ordenada K4 almacena el valor escalar V7 en un segundo subnivel 215 y el valor compuesto V8 en un tercer subnivel 220. V8 contiene dos pares clave-valor escalares, K9 y K10, y sin más valores compuestos identificados, el documento está almacenado en su totalidad.

Con referencia ahora a la Fig. 3, se proporciona un método para mapear los datos de los documentos en forma de uno o más pares clave-valor asociados con el documento en una o más tablas de una estructura tabular de acuerdo con una o más realizaciones. El método 300 descrito asume que el sistema esté en un estado inicial vacío, o sea, sin ninguna tabla de datos o metadatos existente. De todos modos, el método no está limitado a que un sistema esté inicialmente vacío, y puede ser implementado con sistemas en cualquier estado previo. De este modo, el método puede ser utilizado en sistemas persistentes y durables en los cuales la información y los metadatos persisten después de un reinicio o de un fallo del sistema. Además, los metadatos de los cuales se tiene referencia pueden ser utilizados más adelante para reconstruir los datos del documento original y/o reutilizarlos así para acceder a los datos.

El método 300 empieza en el paso 302, en el cual uno o más documentos son proporcionados para el procesado. En una o más realizaciones, los documentos son obtenidos desde la fuente de datos, como una base de datos (por ejemplo, una fuente de datos 140). Por ejemplo, los documentos pueden ser uno o más ficheros JSON o XML. En otras realizaciones, los documentos son proporcionados localmente, como en la memoria 115. Por ejemplo, un usuario puede

introducir información en un almacenamiento de documentos local en el dispositivo de procesamiento de datos 105 a través de un dispositivo de entrada, como un teclado. En el paso 304, un documento en particular es seleccionado como entrada. Por ejemplo, un documento puede ser seleccionado a través de un filtrado, así como es hecho por aquellos con capacidades ordinarias en el arte, o a través de otro proceso de selección (por ejemplo, el primer documento en entrar es el primero en salir o "first in/first out", el primer documento en entrar es el último en salir o "first in/last out", orden alfabético, etc.). El documento seleccionado es enviado a un lector de documentos, paso 306. El lector de documentos lee los documentos para identificar los datos almacenados en el documento, procesarlos y almacenarlos en tablas. Por ejemplo, el lector de documentos identifica los pares clave-valor almacenados dentro del documento en el nivel raíz. El lector de documentos puede ser cualquier dispositivo de lectura de datos, como un aparato de procesamiento de datos 105.

En el paso 308, los datos del documento identificado son analizados para determinar si los datos incluyen valores compuestos. En una o más implementaciones, el dato del valor compuesto es almacenado en una estructura de datos temporal. Por ejemplo, si el lector de documentos identifica un par clave-valor que tiene un valor que es un documento o una agrupación ordenada anidada, el valor es enviado a una estructura de datos temporal, como una pila, hasta que todos los pares clave-valor que tienen valores escalares en un cierto nivel del documento son procesados. La estructura de datos temporal no está limitada a la estructura de pila, esta puede incluir otras estructuras de datos elementales como una lista enlazada, una lista doblemente enlazada, una agrupación ordenada, una cola, etc. Una estructura de pila no es preferida, pero es una realización válida como aquí se describe. El análisis de datos de documentos, de todos modos (y los datos del proceso, como en los siguientes pasos), no requieren una estructura de datos temporal. En otras realizaciones, se implementan técnicas programáticas para determinar los datos de valores compuestos. Técnicas de programación de ejemplo que recaen en el ámbito de las realizaciones de la presente invención incluyen, pero no están limitadas a, la recursividad.

Continuando con la Fig. 3, el método 300 procesa valores no compuestos de los datos de documentos en el nivel actual del documento para generar y poblar las estructuras tabulares. Valores no compuestos (por ejemplo, datos de valores escalares) son procesados primero como un documento y pueden tener pares clave-valor almacenados en múltiples niveles anidados después del nivel raíz (por ejemplo, el nivel base o no anidado). De este modo, procesar todos

los valores escalares de los datos antes de pasar al próximo subnivel asegura que ningún par clave-valor sea omitido en el procesado. En una o más realizaciones, el método implementa estructuras tabulares existentes, genera estructuras tabulares, o implementa una combinación de estructuras tabulares existentes y generadas dependiendo de si el sistema de almacenamiento de destino (por ejemplo, la fuente de datos 140) ya incluye una referencia a una estructura tabular para aquellos tipos de datos. Para cada valor compuesto (o “subdocumento”) el método genera una referencia a una estructura tabular adicional (o “subestructura tabular” o “subtabla”) que es utilizada para almacenar los datos del subnivel.

5

10 En el paso 310, un extractor de estructura genera una referencia a una tabla. Por ejemplo, el extractor de estructura es el extractor de estructura 125. La referencia a una tabla proporciona acceso de escritura y lectura a la tabla y puede ser una referencia a una tabla existente o generada en vista del nombre de una clave y del tipo de valor del par clave-valor (por ejemplo, metadatos). En una o más realizaciones, las columnas de la tabla se asocian con los metadatos.

15 El extractor de estructuras también genera una fila de datos para almacenar los datos en la tabla, en la cual una fila es mapeada a unas específicas columnas de la tabla. Luego, el método 300 continúa y el extractor de datos extrae el valor de un par clave-valor, paso 312. Entonces, el método 300 almacena el valor del dato extraído en la tabla, paso 314, y se referencia el valor con un conjunto de metadatos correspondientes, paso 316. Por ejemplo, cada par clave-valor único

20 (por ejemplo, un valor escalar) en un nivel del documento, es procesado y almacenado en una fila y columna particular de acuerdo con la referencia en los metadatos en la tabla de metadatos (por ejemplo, el tipo y el nombre de un par clave-valor). El método 300 entonces itera para cada valor escalar en el nivel actual del documento. Si el nivel actual del documento identifica datos de un valor compuesto, el método 300 procesa el valor compuesto enviándolo a una estructura de datos temporal (por ejemplo, una pila), como en el paso 308. Después de que todos los valores escalares de los pares clave-valor hayan sido procesados, el método salta a un subnivel de la ruta del documento relativo a un identificador de un valor compuesto de un par clave-valor en la estructura de datos temporal y ejecuta el método otra vez. Esto puede incluir generar y poblar subtablas que estén asociadas con la tabla del nivel raíz.

25

30 Con referencia ahora a la Fig. 4, el flujo del método 400 ilustra el mapeo de uno o más pares clave-valor asociados con un documento en una o más estructuras tabulares de acuerdo con una realización particular. En este flujo del método de ejemplo, un documento que tiene uno o más

pares clave-valor es analizado primero en su nivel raíz, y los valores de los pares clave-valor escalares en el nivel raíz son procesados y almacenados en una estructura tabular. Los valores de los pares clave-valor compuestos son enviados a una estructura de datos temporal hasta que el nivel raíz se complete, y el método 400 salta a los subniveles para procesar los pares clave-valor compuestos en posteriores estructuras subtabulares que tienen una referencia a las estructuras tabulares de nivel más alto con dependencia hasta el nivel raíz.

Más precisamente, el flujo del método 400 empieza en el paso 402 en el cual un dispositivo de procesamiento de datos accede a una fuente de datos que tiene uno o más documentos. Cada documento típicamente tiene uno o más pares clave-valor, aunque el método 400 puede procesar documentos que no tienen datos almacenados. En ese caso, el método 400 o no produce ninguna estructura tabular, o produce una única estructura tabular vacía, dependiendo de la implementación. Además, cada documento tiene uno o más niveles de documento dependiendo si hay subdocumentos anidados (por ejemplo, pares clave-valor compuestos) que son contenidos dentro de los datos del documento. El método 400 es capaz de procesar valores anidados en tantas estructuras subtabulares como niveles anidados existan. En el paso 404, un documento en particular es leído desde la fuente de datos. En una o más implementaciones, el documento es leído por un lector de documentos (por ejemplo, un dispositivo de procesamiento de datos 105). Seguidamente, un par clave-valor es identificado en el nivel raíz del documento, paso 406. El orden de identificación de los pares clave-valor es indiferente, lo cual significa que el método 400 puede procesar primero cualquier par clave-valor en el actual nivel de la ruta en el documento, sin tener en cuenta de la posición donde el par clave-valor es almacenado (por ejemplo, no tiene que procesar el primer par presente en el documento).

Continuando con la referencia a la Fig. 4, en el paso 408, el método 400 genera una estructura tabular del nivel raíz como un valor de la posición de almacenamiento. Por ejemplo, la generación de la estructura tabular incluye preferiblemente la generación de la tabla que tiene una o más columnas y al menos una fila para almacenar los datos del documento. Además, en la generación de la estructura tabular, el nombre de la estructura tabular en el nivel raíz es definido como el identificador de la fuente de datos (por ejemplo, el nombre de la base de datos) o el nombre de la colección donde el documento es almacenado, y al menos una columna de la estructura tabular es definida de acuerdo con el nombre de la clave y el tipo del valor del identificador del par clave-valor. Luego, el valor asociado con el par clave-valor es determinado, paso 410. Por ejemplo, un

aparato de procesamiento de datos puede implementar un código de programa (por ejemplo, un extractor de estructura 125) para determinar cuando el tipo de valor es un valor escalar (por ejemplo, un integer, un double, un boolean, un string, etc.) o un valor compuesto (por ejemplo, una agrupación ordenada, un subdocumento anidado). Si el valor del par clave-valor es determinado como un valor escalar, entonces el método salta al paso 412 y el nombre de la clave es extraído. Por ejemplo, el nombre de la clave es extraído y una columna es generada en la estructura tabular de acuerdo con el método de generación de la columna 500, descrito más abajo. Después, el valor del par clave-valor es almacenado en una fila de la estructura tabular y mapeado con la columna generada, paso 414.

10

De todos modos, si en el paso 410, el tipo del valor asociado con el par clave-valor es determinado como un valor compuesto por el aparato de procesamiento de datos, entonces el método salta al paso 416 y el nombre de la clave es extraído. Por ejemplo, el nombre de la clave es extraído y una columna es generada en la estructura tabular de acuerdo con el método de generación de columna 500, descrito más abajo. Luego, una subestructura tabular asociada con el nombre de la clave extraído es generada, paso 418. Por ejemplo, la subestructura tabular es generada del mismo modo en que se genera una estructura tabular raíz (por ejemplo, como una tabla con columnas y filas), con la excepción de que la subestructura tabular tiene una referencia a la estructura tabular raíz (y cualquier otra estructura tabular que interviene) a través de una columna de referencia a una tabla (o "table_ref"). El nombre de la subestructura tabular es definido, en una o más realizaciones, como el identificador de la fuente de datos o el nombre de la colección, seguido por un guion bajo, seguido por el nombre de la clave extraída. En una o más realizaciones, una columna de referencia es generada tanto en la estructura tabular como en la subestructura tabular que enlaza las dos. Por ejemplo la columna de tipo boolean que enlaza la subestructura tabular como un hijo de la estructura tabular padre es generada, como se muestra en las Fig. 8-12 como se describe en el documento. Dado que los valores compuestos contienen datos almacenados en subdocumentos, para asegurar que todos los datos del documento son procesados y mapeados a una tabla, los valores identificados como valores compuestos son almacenados en una estructura de datos temporal, paso 420. Por ejemplo, la estructura de datos temporal puede ser una estructura de datos elemental, como una pila, una lista enlazada, una lista enlazada doble, una agrupación ordenada, una cola, etc.

30

Cuando el valor del par clave-valor es determinado como un valor escalar o un valor compuesto, el método 400 salta al paso 422, en el cual el método determina cuando hay pares clave-valor adicionales por computar en el nivel de documento actual. De acuerdo con el método 400, antes de avanzar al siguiente nivel de subdocumento, cada uno de los valores escalares de un nivel de documento específico debe ser procesado y almacenado en una estructura tabular, y cada uno de los valores compuestos al mismo nivel del documento debe tener una referencia a la subestructura tabular y los valores anidados deben ser enviados a una estructura de datos temporal. Si hay pares clave-valor adicionales en el actual nivel del documento que aún no hayan sido procesados, entonces el método 400 salta al paso 410. Si no hay más pares clave-valor en el actual nivel del documento, el método salta al paso 424 y determina si hay datos en la estructura de datos temporal. Por ejemplo, si un valor compuesto ha sido identificado en los pasos 410-420, entonces habrá datos en la estructura de datos temporal.

Si el método 400 determina que hay datos en la estructura de datos temporal, el método salta al paso 426 y cambia el valor de la posición de almacenamiento a la subestructura tabular generada. En otras palabras, el método baja de un nivel del documento (por ejemplo, desde el nivel raíz al nivel anidado 1, o desde el nivel anidado 1 al nivel anidado 2, etc.) para poder almacenar los datos anidados. A partir de este punto, el método entra en un bucle en el paso 410 y procesa el siguiente nivel de datos. De este modo, el método 400 proporciona un método recursivo para atravesar un documento que tiene una fuente de pares clave-valor. Si, en el paso 424, el método 400 determina que no hay más datos en la estructura de datos temporal, el método salta al paso 428 y termina. Por ejemplo, no hay más datos en la estructuras de datos temporal cuando un documento no tiene pares clave-valor compuestos, o si todos los pares clave-valor compuestos en el documento han sido procesados.

Con referencia ahora a la Fig. 5, un flujo de un programa de generación de columnas 500 ilustra la generación de columnas de una tabla para almacenar datos de documentos de acuerdo a una o más realizaciones de este documento. Aunque la presente invención permite la extracción de datos de documentos en un sistema con un estado tabular preexistente, la realización actual parte de un estado inicial vacío. En los casos donde actualmente no existan tablas para los datos del documento, las columnas para las tablas de los datos son creadas y se crea una referencia a los datos del documento en los metadatos mientras se extrae la estructura (por ejemplo, el extractor de estructuras 125, el paso 310 del método 300). El método de generación de columna 500

empieza en el paso 502 en el cual el nombre de la clave y el tipo de dato del par clave-valor son extraídos. La extracción del tipo de datos incluye la extracción del nombre de la clave, el valor y el tipo del valor (por ejemplo, single, double, string, boolean, etc.) de un par clave-valor. Por ejemplo, en el paso 502, si el par clave-valor que es mapeado tiene un nombre de clave “pizza” y el valor en un string que contiene “pepperoni”, “pizza” es extraído para generar la columna, “pepperoni” es extraído para ser almacenado en una fila de la tabla, y “_s” es extraído para la columna generada. Otros tipos de datos tienen distintas extensiones que dependen del tipo de valor, como “_i” para integer, “_d” para double, “_b” para booleans, etc.

10 Luego, en el paso 504, el nombre de la columna es calculado. Por ejemplo, continuando con el ejemplo de la pizza, una columna con nombre “pizza_s” es generado. El método 500 entonces determina cuando una columna generada que tiene el mismo nombre calculado existe ya, paso 506. Si la columna no existe, el método 500 crea una columna con el nombre generado, paso 508. Si la columna existe, el método 500 termina, en el paso 510. A partir de este punto, los datos extraídos son almacenados en la columna. Por ejemplo, “pepperoni” se almacenaría en una fila de datos y se mapearía con “pizza_s” en el ejemplo anterior. Además, en una o más realizaciones, si el par clave-valor que se ha extraído está en el nivel raíz del documento, el nombre de la tabla será el nombre de la colección o cualquier otro mecanismo equivalente de nombrado es proporcionado al conjunto de los documentos relacionados. Por ejemplo, si la colección que almacena el par clave valor de ejemplo “pizza” se llama “foods”, entonces el nombre de la tabla será “foods”.

Si un documento tiene un par clave-valor que tiene un nombre de clave idéntico, pero con un tipo de valor en conflicto, entonces el método para generar la estructura tabular emprende unos pasos específicos en una o más realizaciones. Tal conflicto es un problema común que se encuentra en los sistemas de bases de datos documentales convencionales. Dos ejemplos de tal caso son ilustrados en la Fig. 6. En el primer ejemplo, el documento en la entrada incluye un primer par clave-valor 602 que tiene un nombre de clave “key1” y un valor de tipo integer de 33, y un segundo par clave-valor 604 también con un nombre de clave “key1”, pero es un tipo string que tiene un valor “foo”. Entonces, el método de generación de columna 500 genera una primera columna 606 “key1_i” que corresponde al primer par clave-valor 602, y una segunda columna 608 “key1_s” que corresponde al segundo par clave-valor 604. En este caso, el conflicto de tipos no requiere pasos adicionales más allá de los que hay en el método 500, dado que el método genera dos

columnas con nombres distintos, a pesar de que cada par clave-valor tiene el mismo nombre de clave. En el segundo ejemplo, un primer conjunto de pares clave-valor 610 incluye un primer par clave-valor 612 con un nombre de clave "name" y un valor "Joe" de tipo string, y un segundo par clave-valor 614 que tiene un nombre de clave "age" y un valor 35 de tipo integer. Un segundo conjunto de pares clave-valor 616 incluye un tercer par clave-valor 618 con un nombre de clave "name" y un valor "Eve" de tipo string y un cuarto par clave-valor 620 con un nombre de clave "age" y un valor "25" de tipo string. Al implementar el método de generación de la columna 500, dado que tanto el primer par clave-valor 612 como el tercer par clave-valor 618 tienen el mismo nombre de clave y tipo de valor, no hay conflictos y una primera columna 622 es generada con una referencia en los metadatos de "name_s" y "Joe" y "Eve" son almacenados en filas de datos en su orden de procesado. Por el contrario, el segundo par clave-valor 614 y el cuarto par clave-valor 618 tienen un conflicto de tipos porque tienen el mismo nombre de clave "age", pero el segundo par clave valor es un integer, mientras que el cuarto par clave-valor es un string. Para remediar esto, se generan dos columnas separadas, una segunda columna 624 que tiene referencia "age_i" y una tercera columna 626 que tiene referencia "age_s". Sus respectivos valores son mapeados a esas columnas, y las filas que tienen un par clave-valor con referencia a esa columna son dejadas vacías, como se muestra en la Fig. 6.

Para efectuar el mapeo en ambas direcciones entre los pares de clave-valor y el sistema tabular, descrito anteriormente, los sistemas y los métodos de la presente aplicación implementan referencias de metadatos. En una o más realizaciones, el almacenamiento de los metadatos se efectúa en una o más tablas de metadatos que tienen la referencia a definiciones de columnas de las tablas de datos que representan los datos del documento. Los metadatos sirven para dos objetivos fundamentales. Primero, dado que los metadatos tienen una referencia a las columnas de las tablas que representa el documento en su forma original, proporcionan un enlace a la estructura del documento original, de este modo permite la reconstrucción del mismo. Segundo, las referencias en los metadatos no dependen de la implementación, lo que quiere decir que pueden ser una referencia a distintas estructuras tabulares o bases de datos relacionales, si están completamente persistidas, parcialmente persistidas, o sin estar persistidas.

Con referencia ahora a la Fig. 7, un ejemplo de conjunto de tablas de metadatos de acuerdo con una o más realizaciones es ilustrado. Este conjunto de tablas de metadatos, de ejemplo, es meramente una forma en la cual los metadatos pueden hacer referencia a unas tablas de datos

de documentos, y no quiere limitar la presente aplicación a ese conjunto concreto de tablas. Una tabla de metadatos de fuentes de datos 702 es proporcionada para almacenar las referencias a las fuentes de datos que contienen un documento dado. Esta incluye dos columnas, una definida como “name”, que es asignada por el usuario a la fuente de datos, y una definida como “identifier”,
5 que es un identificador interno generado por el método de mapeo de datos de este documento. Típicamente, estos dos son lo mismo, aunque la columna “identifier” puede ser distinta en implementaciones que tienen una fuente de datos que es una base de datos persistente, debido a que los nombres de las bases de datos pueden ser accedidos y modificados en las bases de datos persistentes. En una o más realizaciones, la columna “identifier” es generada para
10 solucionar las limitaciones en el acceso a los datos (“backend”) inherentes a la fuente de datos. Por ejemplo, la columna “identifier” es generada si el backend no permite un cierto número como identificador o tiene unas limitaciones de tamaño. Una tabla de metadatos de colecciones 704 es proporcionada cuando la fuente de datos es una base de datos que tiene una o más colecciones para permitir almacenar las referencias a una colección en particular que contiene un documento
15 dado en la base de datos. La tabla de metadatos de colecciones 704 incluye las mismas columnas de la tabla de metadatos de las fuentes de datos 702, y además añade una tercera columna “database” que referencia la tabla de metadatos de las fuentes de datos.

Al mismo tiempo que un documento es procesado, por ejemplo por el método 300 o el método
20 400, su nivel de estructura y sus subdocumentos (por ejemplo, los valores compuestos, los documentos anidados) se guardan y se mapean a una estructura de tablas en particular. Una tabla de los metadatos de los índices de los documentos 706, es proporcionada para almacenar las referencias a los datos almacenados en las rutas de los documentos y de los subdocumentos contenidos. Una ruta de un documento es una lista ordenada, que empieza desde el nivel raíz,
25 de las claves que hay que atravesar para llegar a un determinado valor. Por ejemplo, la tabla de los metadatos de los índices de los documentos 706, incluye una columna “database” como en el caso anterior, y una columna “collection” que referencia la tabla de metadatos de las colecciones 704, una columna “table_ref” que referencia la ruta a un documento, una columna “identifier” que referencia el identificador interno a la ruta del documento y coincide con al menos
30 una tabla de datos generada, y una columna “last_rid” que referencia la última fila en la tabla de datos generada identificada.

Cada vez que un par clave-valor en un documento es procesado, una referencia entre la clave del documento original y el identificador almacenado por el sistema es almacenada. Por ejemplo, esta referencia es almacenada en una tabla de metadatos de claves 708. La tabla de metadatos de claves 708 puede incluir una columna “name” que es una referencia al nombre de la clave del par clave-valor en el documento (y no el nombre de la fuente de datos, ya que este se referencia en la columna “database”), una columna “type” que referencia el tipo del dato del par clave-valor (sea escalar o compuesto), y una columna “identifier” que referencia el identificador interno de la clave y coincide con una fila en la tabla de los metadatos de los índices de los documentos 706 por las columnas “database”, “collection” y “table_ref”.

10

Debido a que los documentos pueden incluir valores escalares y valores compuestos, para un par clave-valor que tiene una agrupación ordenada, una tabla de metadatos de agrupación ordenada 710 puede ser incluida para tener en cuenta los pares clave-valor que contienen una agrupación ordenada de valores escalares. La tabla de metadatos de agrupación ordenada 710 puede incluir una referencia a columnas por las columnas “database”, “collection” y “table_ref” como en el caso anterior. Además, la tabla de metadatos de agrupación ordenada 710 puede incluir una columna “type” que indica el tipo de dato almacenado en la agrupación ordenada, incluida una referencia para cada fila en la tabla desde la agrupación ordenada, y una columna “identifier” que referencia el nombre interno de la columna que almacena el valor en la tabla relacionada con la ruta en el documento.

20

Además, la Fig. 7 adicionalmente ilustra cuatro columnas generadas automáticamente 712 para las estructuras tabulares de los datos de acuerdo con una o más realizaciones. Estas columnas generadas automáticamente 712 se consideran metadatos y almacenan la relación entre todos los valores de un mismo documento. Por ejemplo, como se ilustra en la Fig. 7, la columna “did” es un identificador de documento que identifica únicamente el documento y todas las filas de cada una de las tablas que pertenecen al mismo documento comparten el mismo valor del “did”; la columna “rid” identifica las filas de una misma tabla para distinguir los casos en los cuales una tabla contiene más de una fila del mismo identificador de documento (por ejemplo, una agrupación ordenada); la columna “pid” es un identificador del padre que referencia la fila padre (por ejemplo, la fila de la cual dependen los pares clave-valor anidados); una columna “seq” que referencia el orden de los elementos de una agrupación ordenada (por ejemplo, para reconstruir el documento en su formato original si necesario). De todos modos, dependiendo del nivel de la

30

5 tabla y del documento contenido, no todas las columnas tienen que ser necesarias. Por ejemplo, en el nivel raíz de un documento, únicamente la columna "did" es necesaria, dado que el orden no es importante, puede solo haber una única fila, y ninguna fila padre de la cual depende. Del mismo modo, un subdocumento de segundo nivel (por ejemplo, un primer documento anidado)

10 únicamente necesita las columnas "did" y "rid" siempre y cuando el subdocumento no sea una agrupación ordenada. En una o más implementaciones, la presente aplicación efectúa optimizaciones automáticas para generar automáticamente y únicamente las columnas de los datos necesaria. De este modo, se ahorran los recursos de la computadora, como el espacio de memoria y de almacenamiento, lo cual es particularmente ventajoso para bases de datos persistentes que pueden ser modificadas de modo continuado.

15 Con referencia a las Fig. 8-12, un documento de ejemplo 800 que tiene múltiples pares de clave-valor es procesado, y los datos contenidos en los pares clave-valor son mapeados para generar una estructura tabular. Este ejemplo se proporciona con carácter ilustrativo para describir más plenamente los sistemas y los métodos de una o más realizaciones descritas en este documento, aunque no se quiere limitar la aplicación únicamente a este ejemplo.

20 En referencia a la Fig. 8, el documento 800 se recibe desde la fuente de datos para el procesado. El documento 800 contiene datos relativos a la identificación del libro "The Martian" de Andy Weir. Por ejemplo, el documento 800 puede ser un registro almacenado en una base de datos de una librería. En este ejemplo, el documento 800 incluye un nombre 805 "book". El nombre 805 puede ser asignado por el usuario, o asignado automáticamente como una referencia a una colección donde el documento es almacenado, o a la fuente de datos misma. Hay tres pares clave-valor almacenados en el nivel raíz del documento de ejemplo 800: un primer par clave-valor 810 que

25 indica el valor que identifica el libro, un segundo par clave-valor 815 que indica el nombre del libro, y un tercer par clave-valor 820 que indica el autor del libro. El primer par clave-valor 810 tiene una clave "id" y un valor escalar 5370 almacenado como un tipo double. El segundo par clave valor 815 tiene una clave "nombre" y un valor escalar de "The Martian" almacenado como un tipo string. El tercer par clave-valor 820 tiene una clave "author" y un valor compuesto de un subdocumento anidado que contiene un cuarto par clave-valor 825 (que indica el nombre del autor del libro) y un quinto par clave-valor 830 (que indica los otros identificadores de libros que tienen el mismo autor). Dado que el tercer par clave-valor es un valor compuesto, el documento

30 800 almacena el cuarto par clave-valor 825 y el quinto par clave-valor 830 en un subnivel. El

cuarto par clave-valor 825 tiene un valor escalar de “Andy Weir” almacenado como tipo string. El quinto par clave-valor 830 tiene un valor compuesto de una agrupación ordenada, la misma agrupación ordenada contiene tres valores escalares cada uno almacenado en un subnivel posterior.

5

Volviendo ahora a la Fig. 9, la estructura tabular generada y mapeada por el documento 800 de acuerdo con los métodos para extraer los datos del documento y mapear los datos a la estructura tabular como se describe en una o más realizaciones aquí ilustradas (por ejemplo, por el método 300, el método 400). En este documento de ejemplo 800, los cinco pares clave-valor son procesados y mapeados a tres tablas de acuerdo con los datos anidados del documento: una

10 tabla de nivel raíz 910, una tabla de primer subnivel 920 que depende de la tabla del nivel raíz, y una tabla de segundo subnivel 930 que depende de la tabla del primer subnivel. Como se muestra en la Fig. 9, el primer par clave-valor 810 y el segundo par clave-valor 815 almacenados en el nivel raíz del documento 800 son mapeados con la tabla del nivel raíz 910, el tercer par clave-

15 valor 820 es mapeado con la tabla del primer subnivel 920 debido al subdocumento anidado que contiene, y el cuarto par clave-valor 825 y el quinto par clave-valor 830 son mapeados con la tabla del segundo subnivel 930. La información utilizada para poblar los nombres de las columnas y los datos de las filas de la estructura tabular utilizada para proporcionar la estructura y la definición de la tabla es proporcionada por la referencia de los metadatos, como se describe en las Fig. 5-

20 7.

Con referencia ahora a la Fig. 10, la generación de la tabla del nivel raíz 910 y el mapeo del primer par clave-valor 810 y del segundo par clave-valor 815 es ilustrado. Para evitar posibles limitaciones resultantes de la solución de almacenamiento utilizada, un nombre interno es

25 asignado para cada identificador que aparece, y utilizado a través del proceso de mapeo de datos. En una o más realizaciones, la estructura de niveles del documento 800 es procesada para generar una o más tablas de metadatos para proporcionar un índice de la ruta en el documento para los pares clave-valor allí almacenados. Por ejemplo, la referencia entre el nombre original y el nombre interno es almacenada en las tablas de los metadatos. En el ejemplo, una tabla de

30 metadatos de los doc_part 1010 proporciona identificadores de referencia para definir la estructura de la tabla del nivel raíz 910. La tabla del nivel raíz 910 en este ejemplo se llama “book” y se basa en un identificador del nivel raíz, que se selecciona en vista de la colección o de la fuente de datos en la cual el documento 800 está almacenado. La columna “table_ref” de la tabla

de metadatos de los doc_part 1010 en un conjunto vacío dado que la tabla del nivel raíz 910 no depende de ninguna tabla de orden más alto (está en el nivel raíz).

La tabla de metadatos de los campos 1020 define la estructura de una columna adicional en la
 5 tabla del nivel raíz 910 de acuerdo al método de generación de columnas 500. Como se ilustra
 en la Fig. 10, la tabla del nivel raíz 910 referencia la tabla de metadatos de los campos 1020 para
 generar las columnas “id_” (que referencia el primer par clave-valor 810 y el valor de tipo integer),
 “name_s” (que referencia el segundo par clave-valor 815 y el tipo de valor string), y “author_b”
 (que referencia el tercer par clave-valor 820 y el tipo de valor boolean). En otras realizaciones en
 10 las cuales la tabla se mapea a una preexistente (por ejemplo, una tabla del nivel raíz llamada
 “book” que ya existe), estos pasos para la generación de los metadatos de la columnas no son
 necesarios si el tipo del par clave-valor coincide. Desde este punto, una fila de datos es generada
 en la tabla del nivel raíz 910 y es poblada con los valores de los datos de los pares clave-valor
 del nivel raíz (por ejemplo, 5370, “The Martian”, y false (falso) – que quiere decir que este valor
 15 no es escalar). Para los valores compuestos, como el tercer par clave-valor 820, los valores no
 se almacenan en la tabla del nivel raíz 910. En lugar de eso, estos valores se envían a una
 estructura de datos temporal para ser procesado más adelante. El uso de una estructura de datos
 temporal, por ejemplo una pila de operaciones es mostrada en la Fig. 13 y es descrita plenamente
 más adelante. El proceso continúa hasta que cada par clave-valor en el nivel raíz es extraído y
 20 mapeado en la tabla del nivel raíz 910.

Con referencia ahora a la Fig. 11, la generación de la tabla del primer subnivel 920 y el mapeo
 del tercer par clave-valor 820 y del cuarto par clave-valor 825 es ilustrado. La metodología para
 generar la tabla del primer subnivel 920 es similar a aquella para la generación de la tabla del
 25 nivel raíz 910, con la excepción de que las referencias adicionales a los metadatos de las
 columnas son generadas. En el ejemplo, la tabla de metadatos de los doc_part 1010 proporciona
 los identificadores de referencia que definen la estructura de la tabla del primer subnivel 920 (por
 ejemplo, una ruta en el documento). Por ejemplo, la tabla del primer subnivel 920 de este ejemplo
 es llamada “book_author” dado que la columna “table_ref” de la tabla de metadatos de los
 30 doc_part 1010 ahora incluye una referencia a “author”, que es el nombre de la clave del tercer
 par clave-valor 820 y el actual nivel del documento. Como se comenta anteriormente, la tabla de
 metadatos de los campos 1020 define la estructura de la columna adicional de la tabla del primer
 subnivel 910 de acuerdo con el método de generación de la columna 500. Aquí, la tabla del primer

subnivel 920 referencia la tabla de los metadatos de los campos 1020 para generar las columnas “name_s” (que referencia el cuarto par clave-valor 825 y el tipo string), y “books_b” (que referencia el quinto par clave-valor 830 y el tipo boolean). A partir de aquí, una fila de datos es generada en la tabla del primer subnivel 920 y es poblada con los valores de los datos de los pares clave-valor en el nivel de “book_author” (por ejemplo, “Andy Weir” y true (verdadero) – que significa que este valor no es un valor escalar, y tiene únicamente valores escalares anidados en él). El valor compuesto del quinto par clave-valor es movido a una estructura de datos temporal para ser procesado más adelante. Esta estructura de datos temporal puede ser la misma u otra de la estructura de datos temporal que recibe los valores compuestos enviados desde el nivel raíz.

10

Con referencia ahora a la Fig. 12, la generación de la tabla del segundo subnivel 930 y el mapeo del quinto par clave-valor 830 es ilustrada. Así como en el nivel “book_author”, la tabla de metadatos de los doc_part 1010 proporciona un identificador para definir la tabla del segundo subnivel 930. En este caso, el quinto par clave-valor 830 tiene el nombre “books”, que es añadido a la columna “table_ref” de la tabla de metadatos de los doc_part 1010 para generar el nombre de la tabla del segundo subnivel 930 “book_author_books”. El quinto par clave-valor 830 es una agrupación ordenada que representa tres valores escalares y la metodología aquí proporcionada genera una tabla de metadatos escalares 1210 con una columna que representa el tipo del valor del dato double del quinto par clave-valor. Como en la tabla del nivel raíz 910 y en la tabla del primer subnivel 920, una fila de datos es generada para almacenar los valores procesados del quinto par clave-valor 830; de todos modos, dado que hay tres valores escalares, tres filas de datos son generadas y los valores mapeados a la columna “v_d” (por ejemplo, “value” de tipo double). Para mantener la integridad del orden de estos valores, la tabla del segundo subnivel 930 genera una columna “rid” (para identificar la fila) y una columna “seq” (para identificar la secuencia de los valores en la agrupación ordenada). Dado que no hay ningún otro valor compuesto en el documento 800, el proceso termina y las tres tablas de datos creadas son almacenadas en la base de datos o en otro sistema de almacenamiento deseado.

20

25

30

Con referencia ahora a la Fig. 13, una operación de pila de ejemplo 1300 de un procesado de la estructura de datos temporal es ilustrado. En una o más implementaciones, una operación de pila 1300 es implementada para almacenar temporalmente los datos de un valor compuesto mientras los datos de un valor escalar son procesados, y por eso proporciona una mejora en la eficiencia de mapeo de datos y asegura que ningún dato del documento sea omitido. Otras estructuras de

datos temporales son apropiadas para ser utilizadas con realizaciones de la presente invención, como listas enlazadas, listas doblemente enlazadas, colas, agrupaciones ordenadas, y similares. Una operación de pila 1300 implementa una metodología de procesado LIFO (“last in, first out” - lo último que entra es lo primero en salir) en la cual el proceso envía los datos a la pila hasta el momento más apropiado del procesamiento, en el cual los datos se extraen de la pila y se procesan. En una o más realizaciones, la operación de pila 1300 es independiente del orden, lo cual significa que el procesado de los pares clave-valor de un documento puede efectuarse en cualquier orden de acuerdo con el nivel de la ruta en el documento.

La Fig. 13 ilustra la ruta en el documento para el documento 1305 que tiene ocho pares clave-valor que son valores compuestos anidados. En este ejemplo, las claves 1, 3, 4, 6 y 8 son tipos de valores escalares, y las claves 2, 5, y 7 son tipos de valores compuestos. Dado que la operación de la pila 1300 es independiente del orden, la Fig. 13 ilustra dos órdenes de procesado diferentes de acuerdo con una realización particular. Independientemente del orden y del momento en el que se procesen, el resultado del mapeo es idéntico.

En el primer orden 1310 se procesan las clave-valor de acuerdo con la llegada de la ruta en el documento. En el primer orden 1310, la pila está inicialmente vacía. En la primera iteración de la operación de pila 1300, el nivel raíz del documento 1305 es leído, y cada valor escalar es procesado de acuerdo con los métodos de extracción y mapeo aquí descritos. En este caso, el único valor escalar en el nivel raíz es la clave 1. Las claves 2 y 7 son valores compuestos y son enviados a la pila en el mismo orden de llegada, lo cual quiere decir que dado que la clave 2 llega primero en la ruta del documento, será puesta encima de la clave 7 para ser procesada en primer lugar 1310. Luego, la operación de pila efectúa una segunda iteración para procesar los pares clave-valor aquí almacenados. Dado que la operación de pila 1300 es una operación LIFO, el último valor compuesto almacenado en la pila, por ejemplo, la clave 2, es procesada primero. La clave 2 incluye dos valores escalares (clave 3 y clave 4), que son extraídos y mapeados a una tabla de datos. La clave 5 es un subdocumento anidado y es enviado a la cabeza de la pila. De todos modos, ahora la clave 5 es el último par clave-valor, y la segunda iteración interrumpe el procesado de la clave 2 y se itera la operación de pila 1300 una tercera vez para procesar la clave 5. La clave 5 contiene un único escalar, la clave 6, que es procesada. Dado que no hay más pares clave-valor en este nivel, la clave 5 se extrae de la pila y el primer orden 1310 regresa al punto donde se había interrumpido en la segunda iteración. La clave 2 tampoco tiene más pares clave-

valor, y por eso la clave 2 se extrae de la pila. El primer orden 1310 entonces itera una cuarta vez para procesar la clave 7, que tiene una única clave escalar key8. Entonces, sin datos almacenados en la pila, la operación de pila 1300 termina.

- 5 En el segundo orden 1320, se procesa todo un nivel antes de continuar por los valores compuestos almacenados en la pila en dicho nivel y anteriores. Como antes, el segundo orden 1320 empieza con una pila vacía. Tras la primera iteración, la operación de pila 1300 identifica dos valores compuestos, la clave 2 y la clave 7, y los envía a la pila en el orden de llegada, en lugar de su orden en el documento, lo cual quiere decir que la clave 2 es leída primero y enviada a la pila primero, y que la clave 7 queda finalmente en la cabeza de la pila. Tras la segunda iteración, la operación de pila 1300 procesa los datos de la clave 7 y los extrae de la pila. Tras la tercera iteración, la operación de pila 1300 procesa los datos de la clave 2, identifica la clave 5 con un valor compuesto, y envía la clave 5 a la pila. Tras la cuarta iteración, la operación de pila 1300 procesa los datos de la clave 5 y para completar, extrae la estructura de la clave 5, entonces termina el procesado de los datos de la clave 2.

Las figuras de la 1 hasta la 13 son ilustraciones conceptuales que proporcionan una explicación de la presente invención. Los expertos en la materia deberían estar capacitados para entender que los distintos aspectos de las implementaciones de la presente invención pueden ser implementadas en hardware, firmware, software, o combinaciones de esos. En tales implementaciones, los múltiples componentes y/o pasos serían implementados en hardware, firmware, y/o software para realizar las funciones de la presente invención. Lo que quiere decir, que la misma pieza de hardware, firmware, o módulo de software pueden efectuar uno o más de los bloques ilustrados (por ejemplo, componentes o pasos).

25 En las implementaciones software, el software de ordenador (por ejemplo, programas u otras instrucciones) y/o datos son almacenados en un medio que la maquina puede leer como parte de un producto de programa de ordenador, y son cargados en el sistema del ordenador u otro dispositivo o maquina a través de un dispositivo de almacenamiento extraíble, disco duro, o interfaz de comunicación. Los programas de ordenador (también llamados lógica de control de ordenador o código de programa legible por un ordenador) son almacenados en una memoria principal y/o secundaria, y ejecutados por uno o más procesadores (controladores, o similares) para producir que se ejecuten las funciones de la invención como se ha descrito en este

documento. En este documento, el término “medio leíble por ordenador”, “medio de programación de ordenador” y “medio utilizable por ordenador” son utilizados generalmente para referirse a medios como una memoria de acceso aleatorio (Random Access Memory o “RAM”); memoria de solo lectura (Read Only Memory o “ROM”); una unidad de almacenamiento extraíble (por ejemplo, un disco magnético u óptico, un dispositivo de memoria flash, o similares); un disco duro; o similar.

Particularmente, las figuras y ejemplos anteriores no quieren limitar el ámbito de la presente invención a una única implementación, dado que otras implementaciones son posibles si se intercambian algunos o todos los elementos descritos o ilustrados. Además, donde ciertos elementos de la presente invención pueden ser parcial o plenamente implementados utilizando los componentes conocidos, únicamente esas partes son descritas evitando entrar en detalle en favor de la claridad de lectura. En la presente memoria, una implementación que muestra un componente singular no debería necesariamente estar limitado a otras implementaciones que incluyen una pluralidad del mismo componente, y vice-versa, a menos que se exprese explícitamente en este documento. Además, los solicitantes no pretenden que ningún término en el pliego de condiciones o las reivindicaciones se le atribuya un significado especial a menos que se exprese lo contrario. Además, la presente invención abarca los conocimientos presentes y futuros equivalentes a los componentes conocidos de los cuales se hace referencia en este documento a modo de ejemplo.

La anterior descripción de las implementaciones revelará plenamente la naturaleza general de la invención que otros pueden, aplicando los conocimientos en el arte pertinente (o los artes pertinentes) (incluido los contenidos de los documentos citados e incorporados con referencias en este documento), modificar y/o adaptar para varias aplicaciones tales implementaciones específicas, sin exagerar la experimentación, sin separarse del concepto general de la presente invención. Tales adaptaciones y modificaciones son por lo tanto previstas dentro del significado y en el rango de equivalencia de las implementaciones divulgadas, basándose en las enseñanzas y las direcciones presentadas en este documento. Hay que entender que la redacción o la terminología de este documento tiene como objetivo la descripción y no la limitación, de modo que la redacción o terminología de la presente memoria deben ser interpretadas por el experto en la materia a la luz de las enseñanzas y las directrices presentadas en este documento, junto con el conocimiento de un experto en la técnica relevante.

Mientras varias implementaciones de la presente invención han sido descritas anteriormente, hay que entender que han sido presentadas a modo de ejemplo, y no de limitación. Será evidente para un experto de la técnica relevante que se pueden hacer varios cambios en formas y detalles sin separarse del espíritu y del ámbito de la invención. Por lo tanto, la presente invención no debería ser limitada por ninguna de las implementaciones de ejemplo descrita anteriormente, en vez de eso debería ser definido únicamente de acuerdo con las siguientes reivindicaciones y sus equivalentes.

REIVINDICACIONES

1. Método de mapeo de uno o más pares clave-valor asociados con un documento en una o más estructuras tabulares, cada uno del o de los pares clave-valor tiene un nombre de clave y un valor, y cada una de la o de las estructuras tabulares tiene una o más filas y columnas para almacenar los valores, comprendiendo el método:
- 5 leer el documento, por un lector de documento, para identificar el o los pares clave-valor asociados con el documento de entrada; determinar cuándo un valor asociado a un dato de los uno o más pares clave-valor asociado con el documento de entrada; en el caso de que el valor
- 10 asociado con el par clave-valor sea un valor escalar:
- extraer el nombre de la clave de par clave-valor, almacenar el valor del par clave-valor en una fila de la estructura tabular, o en el caso de que el par clave-valor sea un valor compuesto:
- 15 extraer el nombre de la clave del par clave-valor; generar una subestructura tabular asociada con el nombre de la clave extraída del par clave-valor.
2. Método de acuerdo a la reivindicación 1, que además comprende:
- en el caso de que el valor asociado con un par clave-valor sea un valor compuesto:
- enviar el valor del par clave-valor a una estructura de datos temporal.
- 20 3. Método de acuerdo a la reivindicación 2, que además comprende:
- determinar cuando el valor compuesto asociado con un par clave-valor que es enviado a la estructura de datos temporal incluye uno o más subpares clave-valor; determinar cuando los uno o más subpares clave-valor son valores escalares o un valor compuesto para cada uno de los uno o más subpares clave-valor; en el caso de que el valor asociado con un subpar clave-
- 25 valor sea un valor escalar:
- extraer el nombre de la clave del subpar clave-valor, comprobar cuando una subestructura tabular tiene una columna asociada con el nombre de la clave extraída del subpar clave-valor, y si es así, almacenar el valor de subpar clave-valor en una fila de la subestructura tabular y mapear el valor del subpar clave-valor a la columna, o si no es así, generar una
- 30 nueva columna asociada con la subestructura tabular, almacenando el valor del subpar clave-valor en una fila de la subestructura tabular y mapear el valor con un subpar clave-valor en la nueva columna; en el caso de que el valor asociado con el subpar clave-valor sea un valor compuesto:

extraer el nombre de la clave del subpar clave-valor; generar una nueva subestructura tabular asociada con el nombre de la clave extraído del subpar clave-valor; y enviar el valor del subpar clave-valor a una estructura de datos temporal:

5 en el caso de que ningún valor asociado con el subpar clave-valor sea un valor compuesto, eliminar el par clave-valor de la estructura de datos temporal.

10 4. Método de acuerdo a la reivindicación 3, que comprende además iterar los pasos de la reivindicación 3 hasta que la estructura de datos temporal no contenga ningún par clave-valor enviado.

5 5. Método de acuerdo a la reivindicación 1, que comprende además en el caso de que el valor asociado con el par-clave valor sea un valor escalar, comprobar que la estructura tabular tenga una columna asociada con el nombre de la clave extraído del par clave-valor.

15 6. Método de acuerdo a la reivindicación 5, que comprende además la generación de una nueva columna asociada con la estructura tabular y que almacena el valor del par clave-valor en una fila de la estructura tabular.

20 7. Método de acuerdo a la reivindicación 5, donde la columna es identificada por un tipo asociado con el valor del par clave-valor.

8. Método de acuerdo a la reivindicación 1, donde el documento tiene formato JSON.

25 9. Método de acuerdo a la reivindicación 1, donde el documento tiene formato XML.

10 10. Método de acuerdo a la reivindicación 1, donde la estructura tabular es un almacenamiento persistente.

30 11. Método de acuerdo a la reivindicación 1, donde el documento entra desde una fuente de datos.

12. Método de acuerdo a la reivindicación 1, donde la fuente de datos es una base de datos relacional.

13. Método de acuerdo a la reivindicación 1, donde la estructura de datos temporal es una pila.

5 14. Método de acuerdo a la reivindicación 1, donde la estructura de datos temporal es una lista enlazada.

15. Método de acuerdo a la reivindicación 1, que además comprende:

10 en el caso de que el valor asociado con el par clave-valor sea un valor compuesto: iterar recursivamente los pasos de la reivindicación 1 hasta que cada uno de los uno o más pares clave-valor sean extraídos a una estructura tabular o subestructura tabular.

16. Un sistema para extraer y almacenar los datos de un documento en una base de datos, donde el sistema comprende:

15 un aparato de procesamiento de datos que incluye un procesador y una memoria acoplada con el procesador;

una fuente de datos que contiene uno o más documentos, donde cada documento tiene uno o más pares clave-valor, cada par clave-valor tiene un nombre de clave y un valor, cada valor tiene un tipo de valor;

20 un lector de documentos para recibir el documento desde la fuente de datos a través de una red, el lector de documentos está acoplado para comunicar al aparato de procesamiento de datos;

25 un módulo para la extracción de estructuras que implementa un código de programa por el procesador para generar una o más estructuras tabulares que tienen al menos una columna que corresponde al nombre de la clave y al tipo del valor de cada uno de los pares clave-valor;

un módulo para la extracción de datos que implementa un código de programa para el procesador para extraer el valor de una fila de datos en una o más estructuras tabulares creada por el módulo de extracción de estructuras; y

30 uno o más tablas de metadatos generadas por el módulo de extracción de estructuras con referencia a una o más estructuras tabulares.

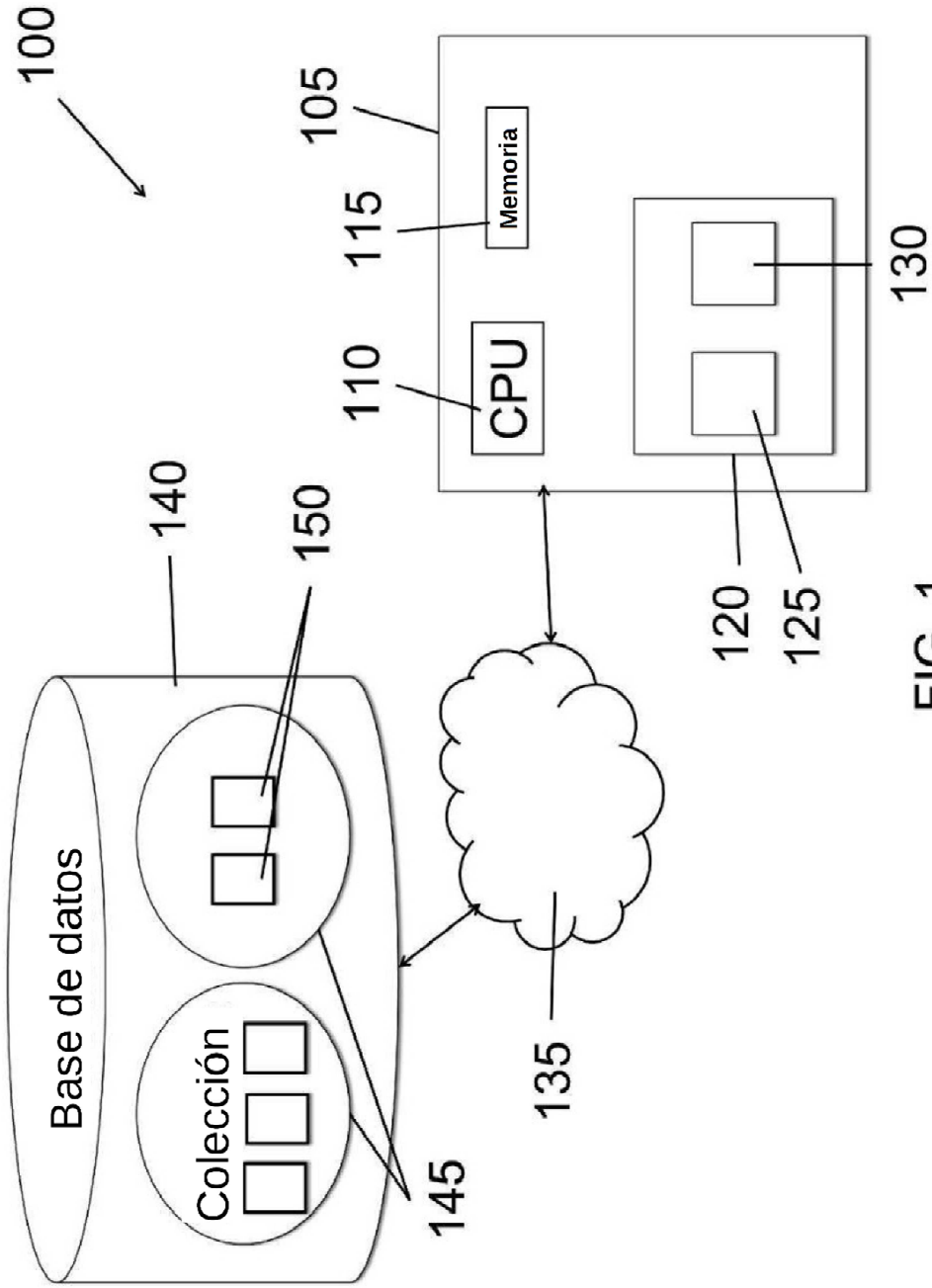


FIG. 1

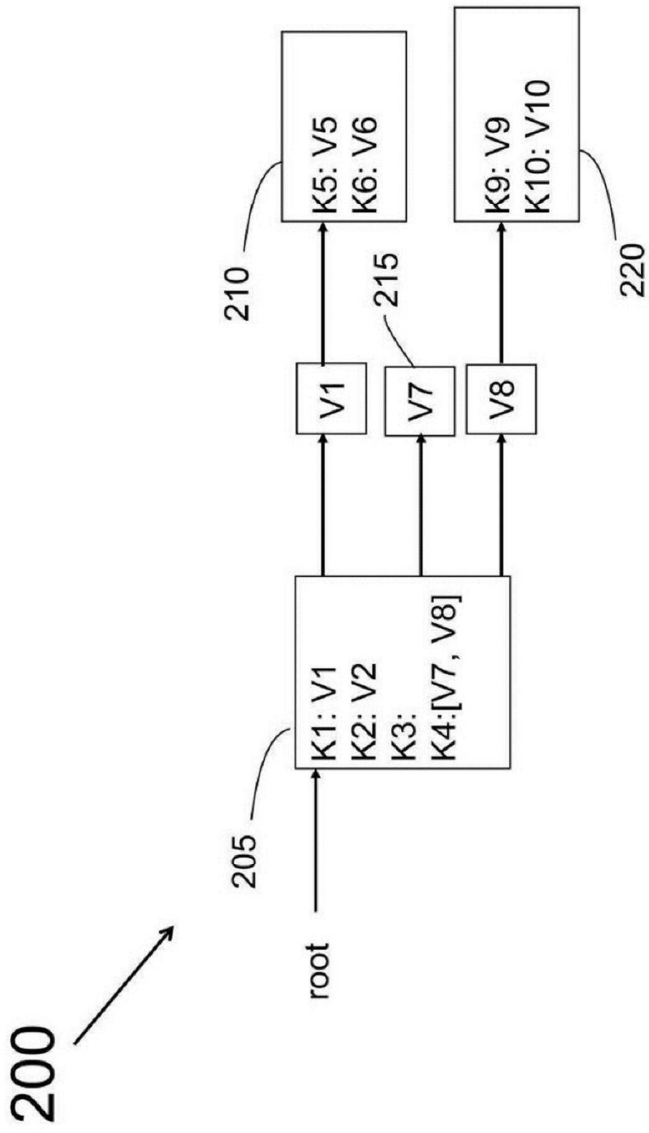


FIG. 2

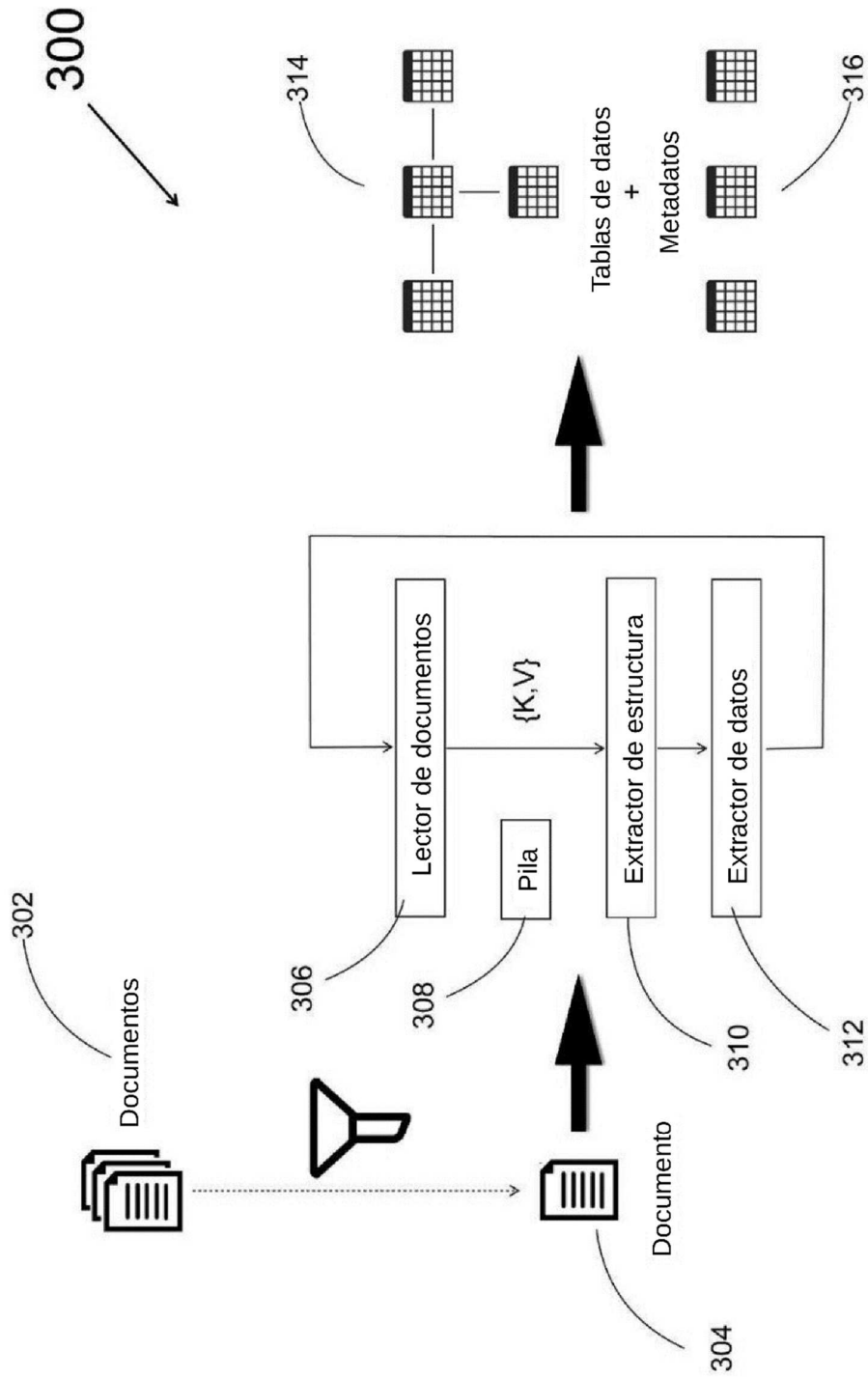


FIG. 3

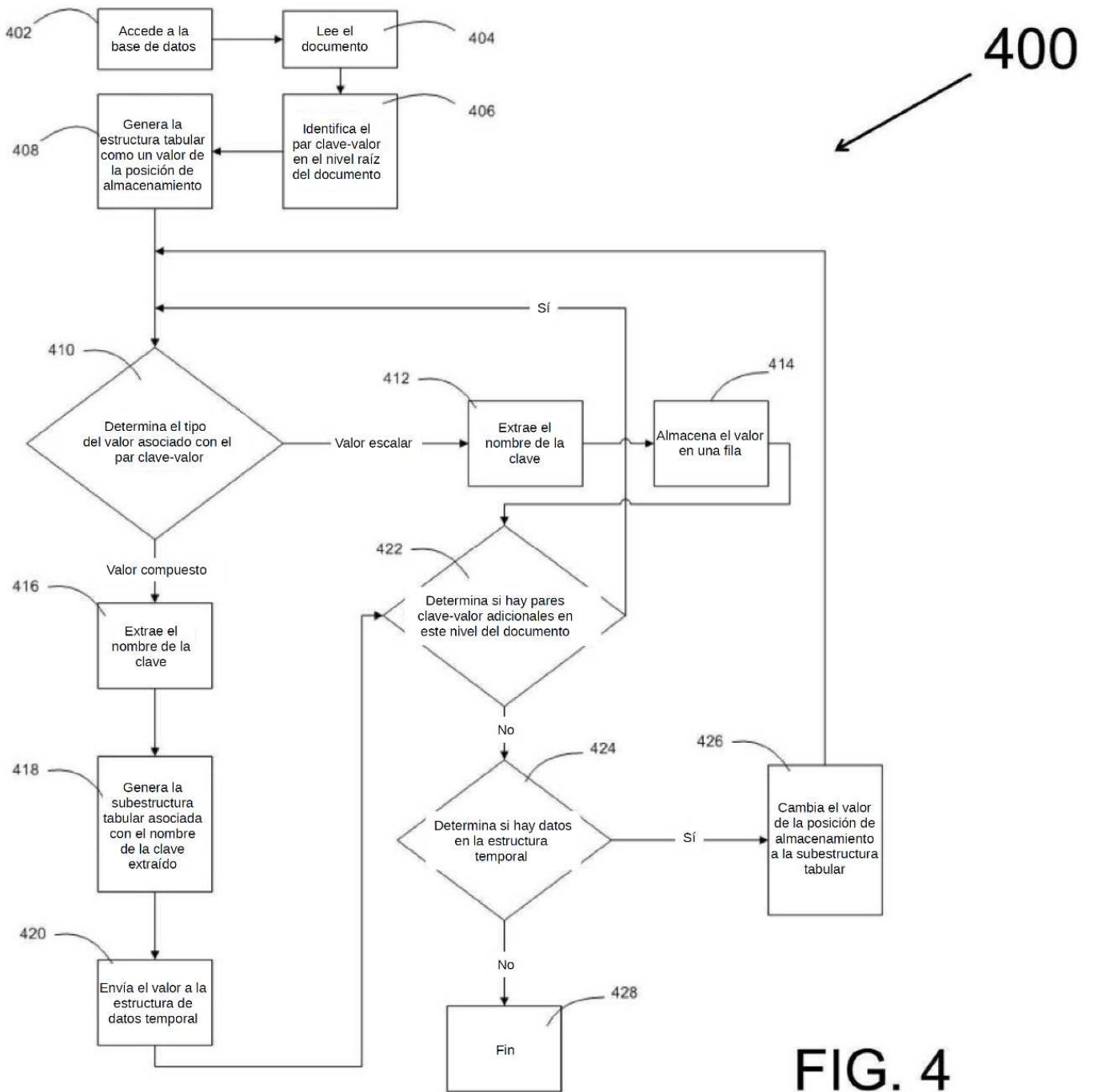


FIG. 4

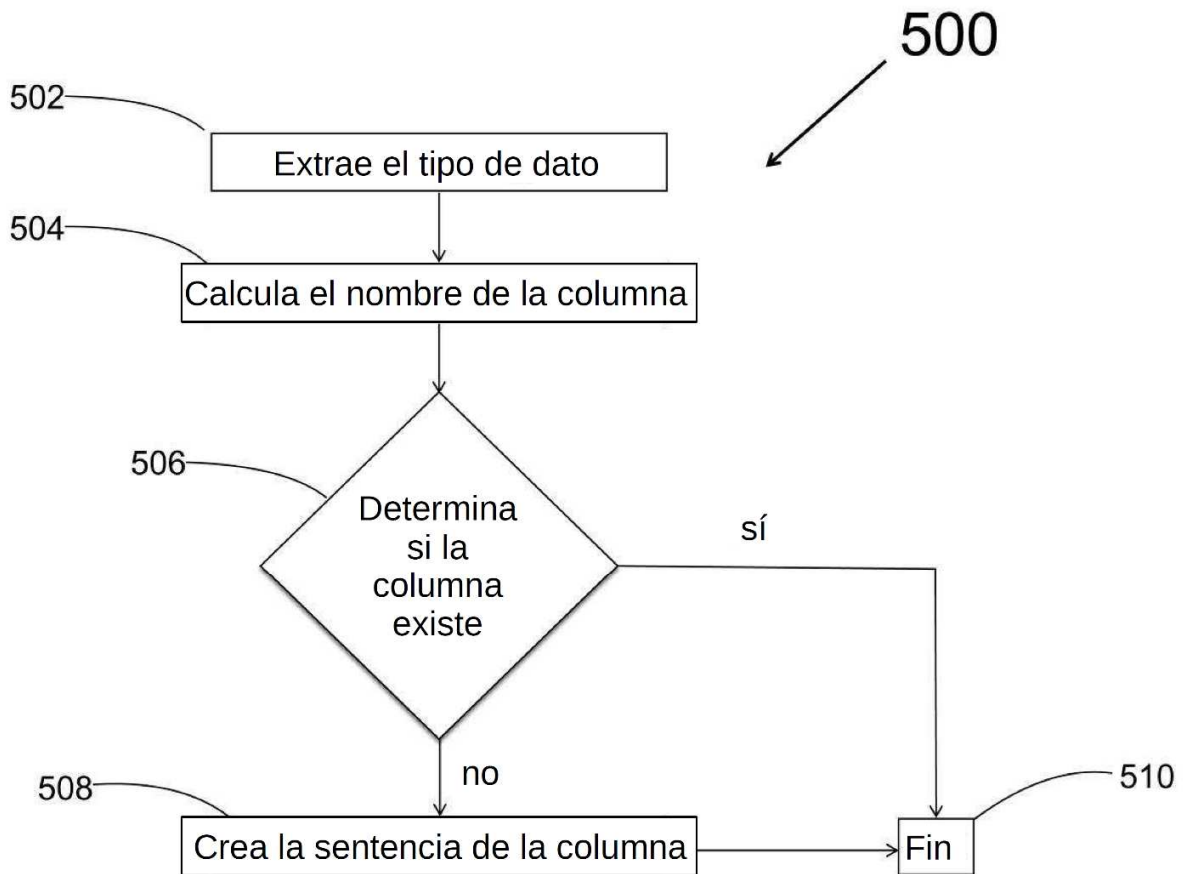


FIG. 5

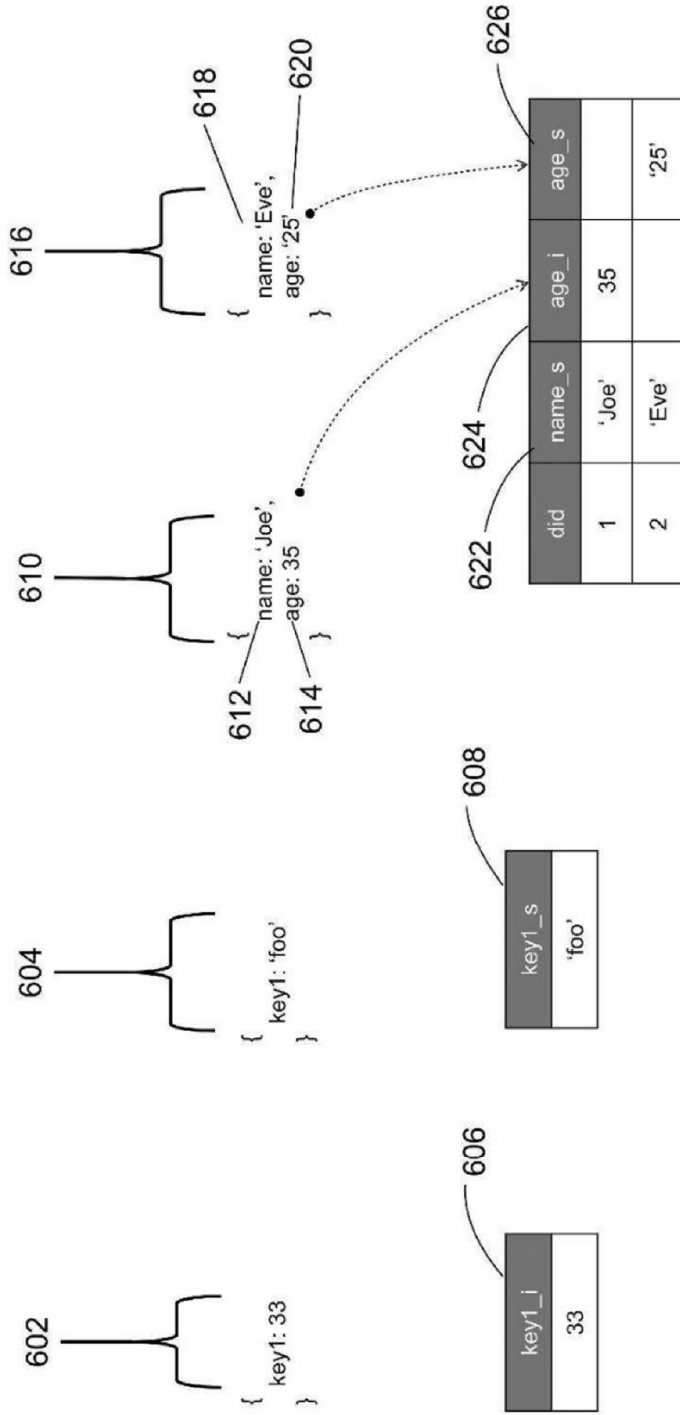


FIG. 6

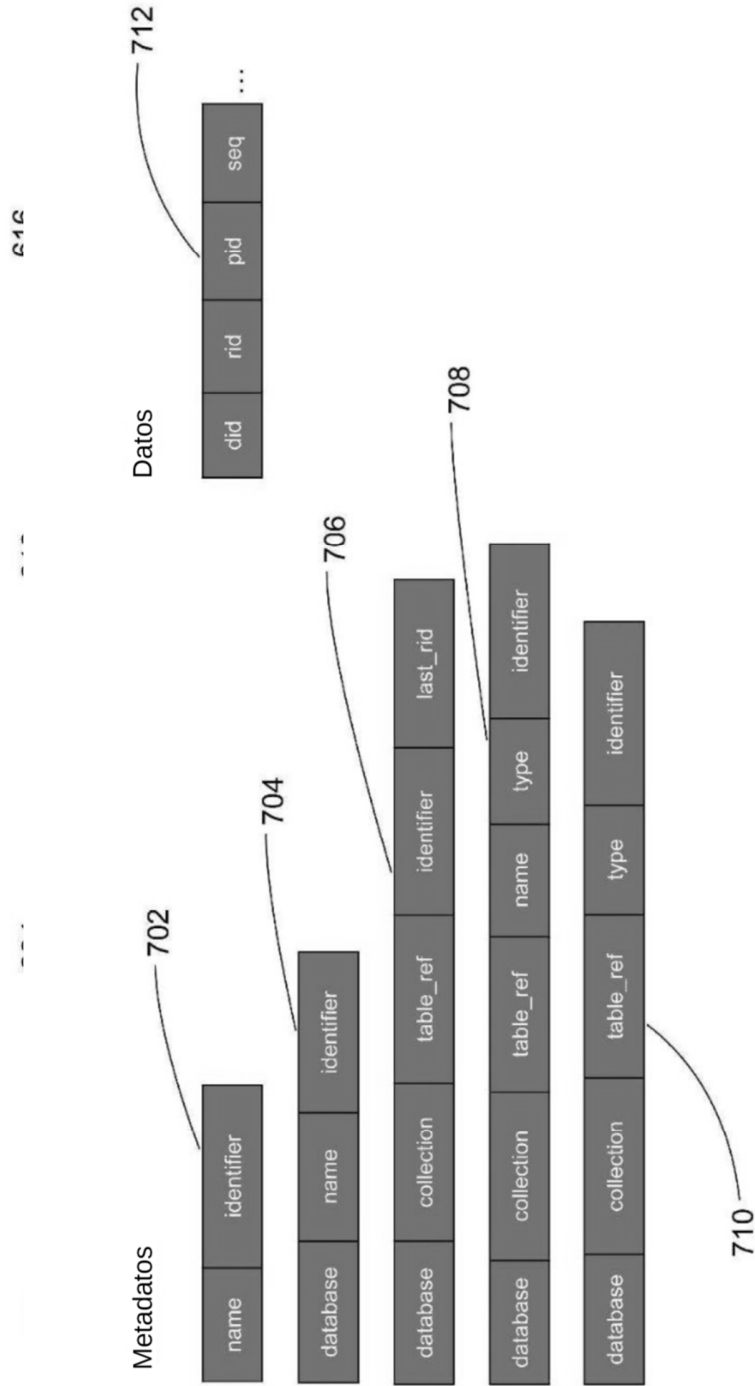


FIG. 7

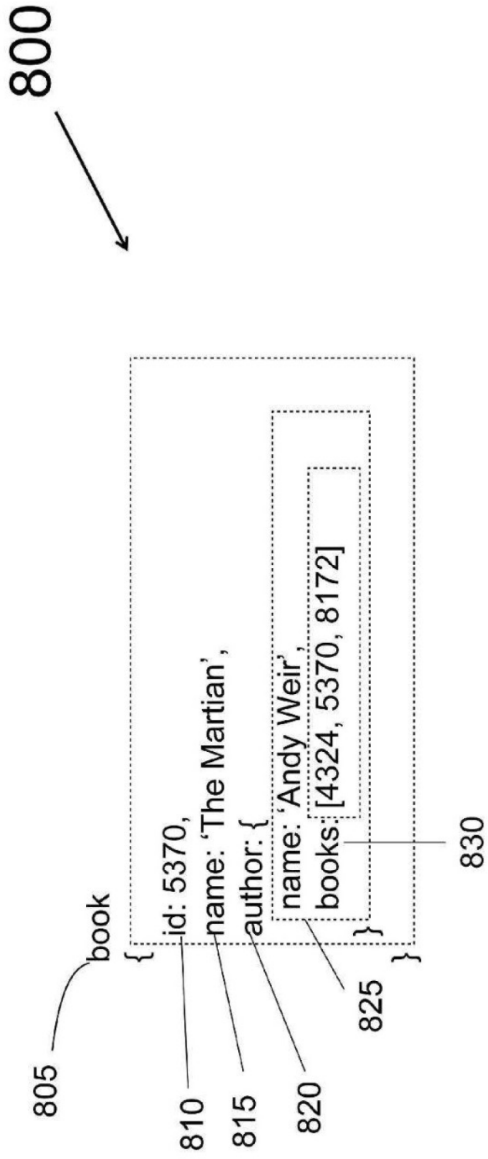


FIG. 8

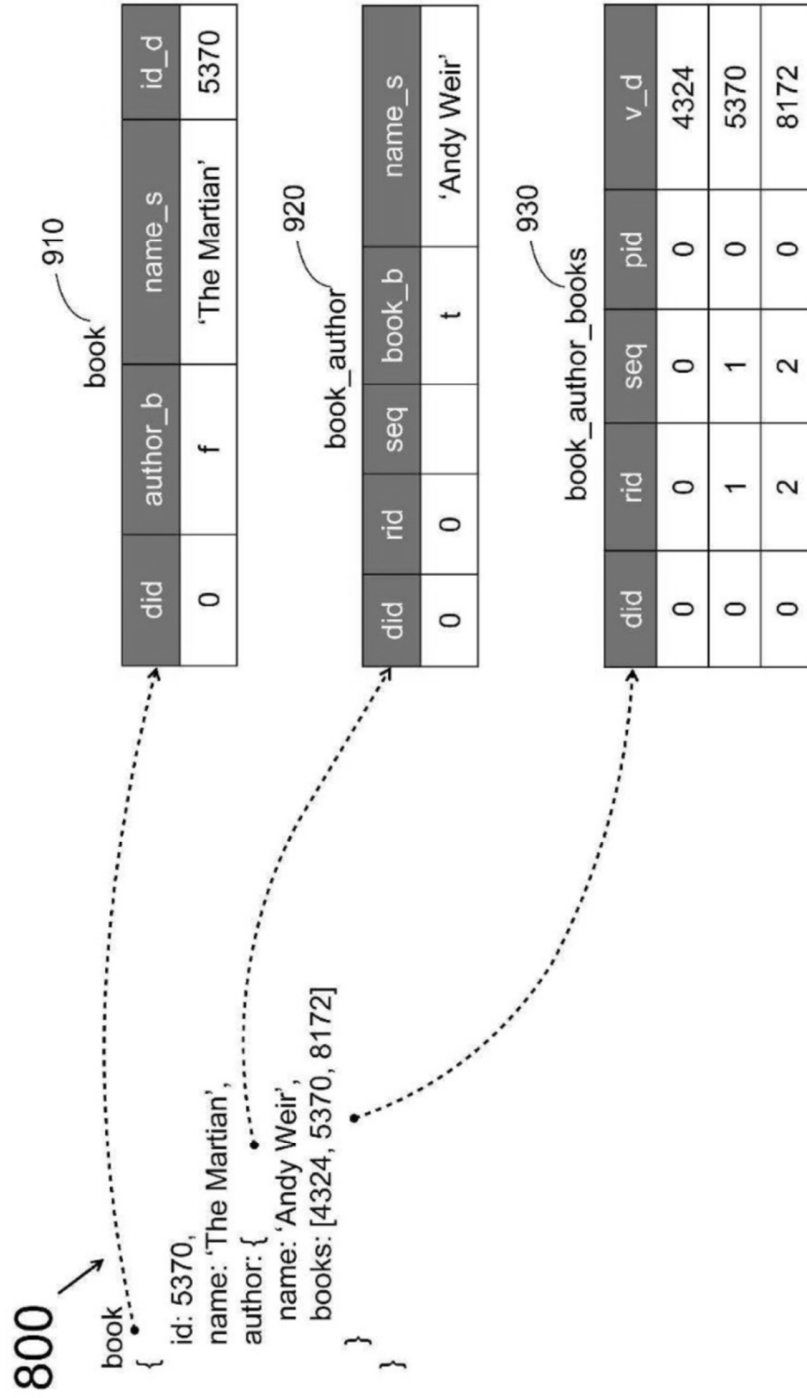


FIG. 9

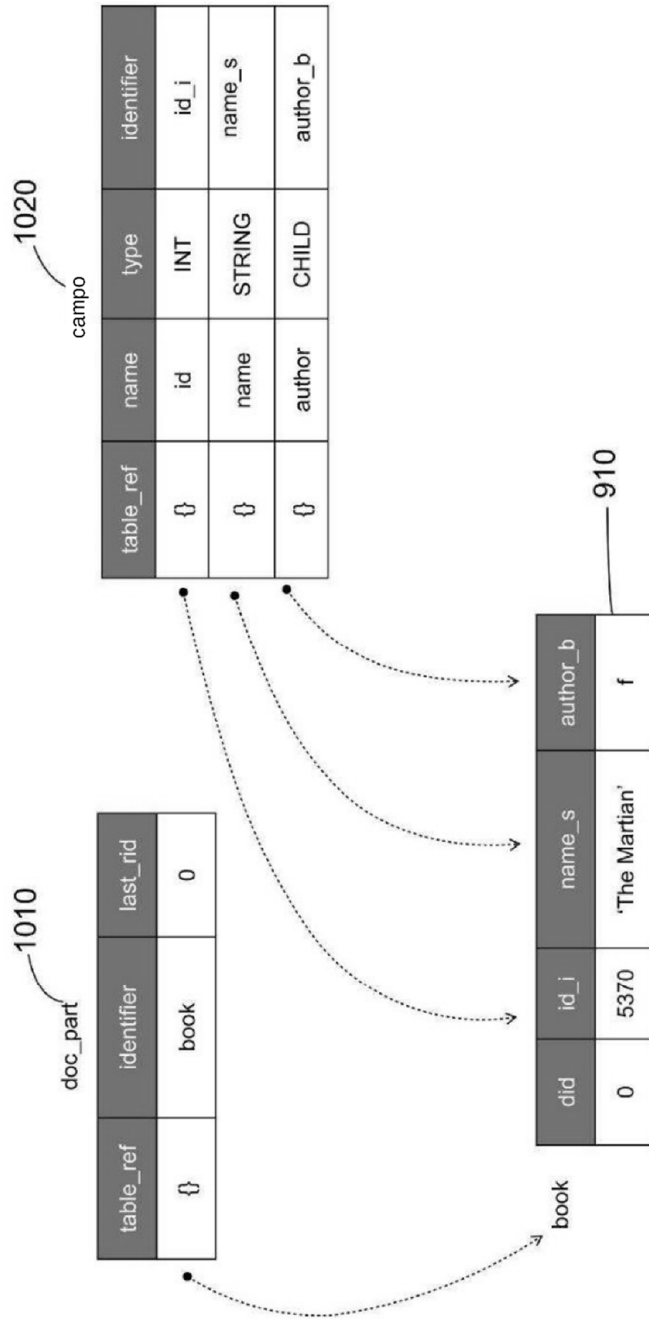


FIG. 10

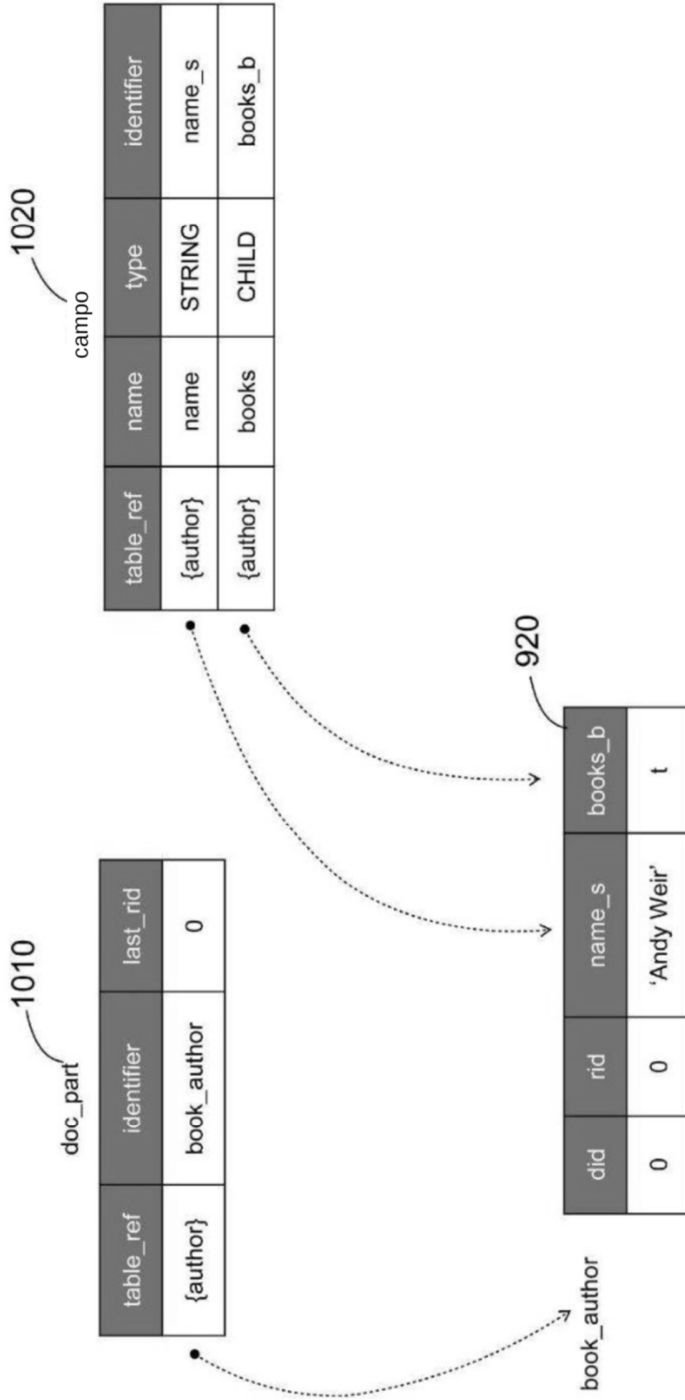


FIG. 11

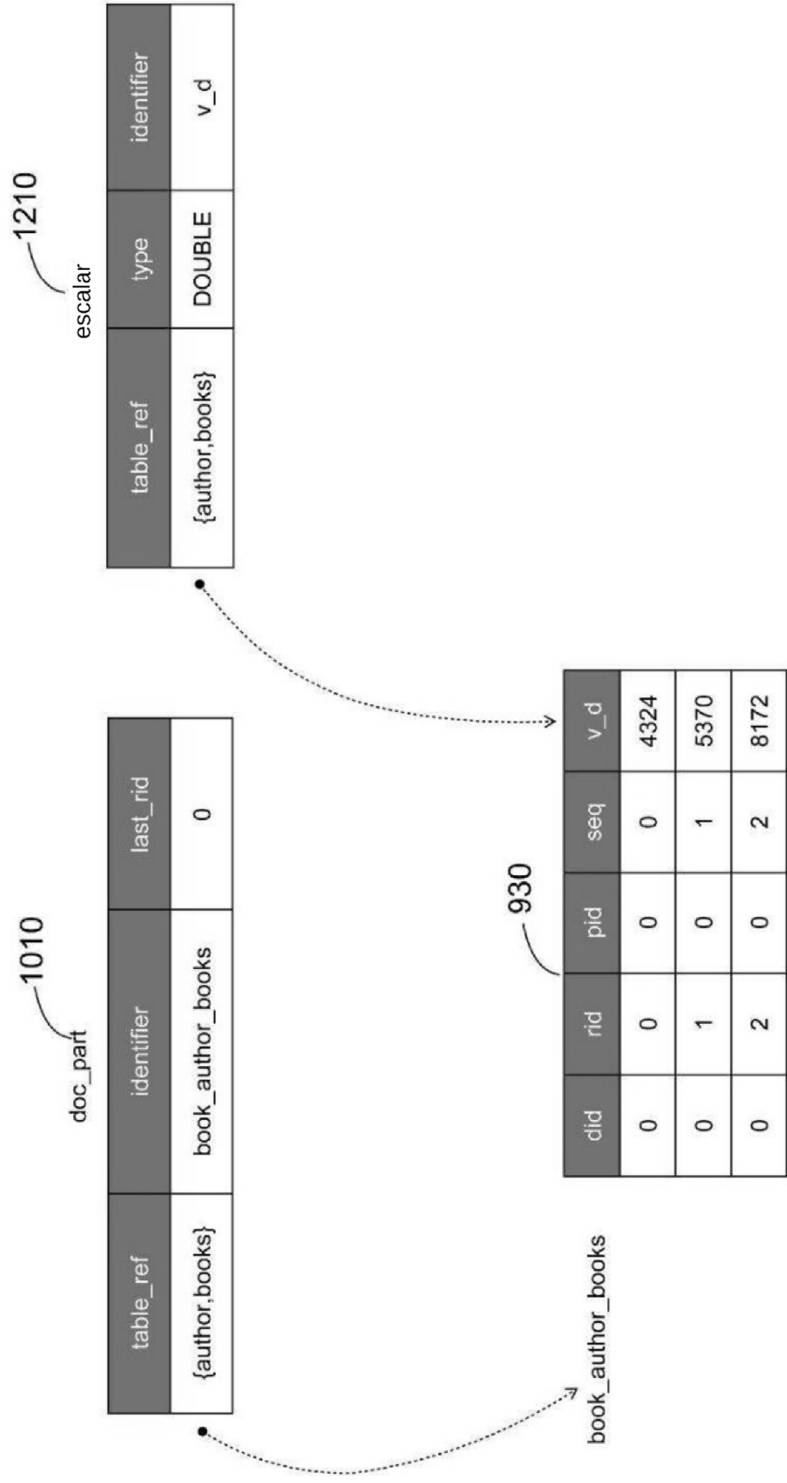


FIG. 12

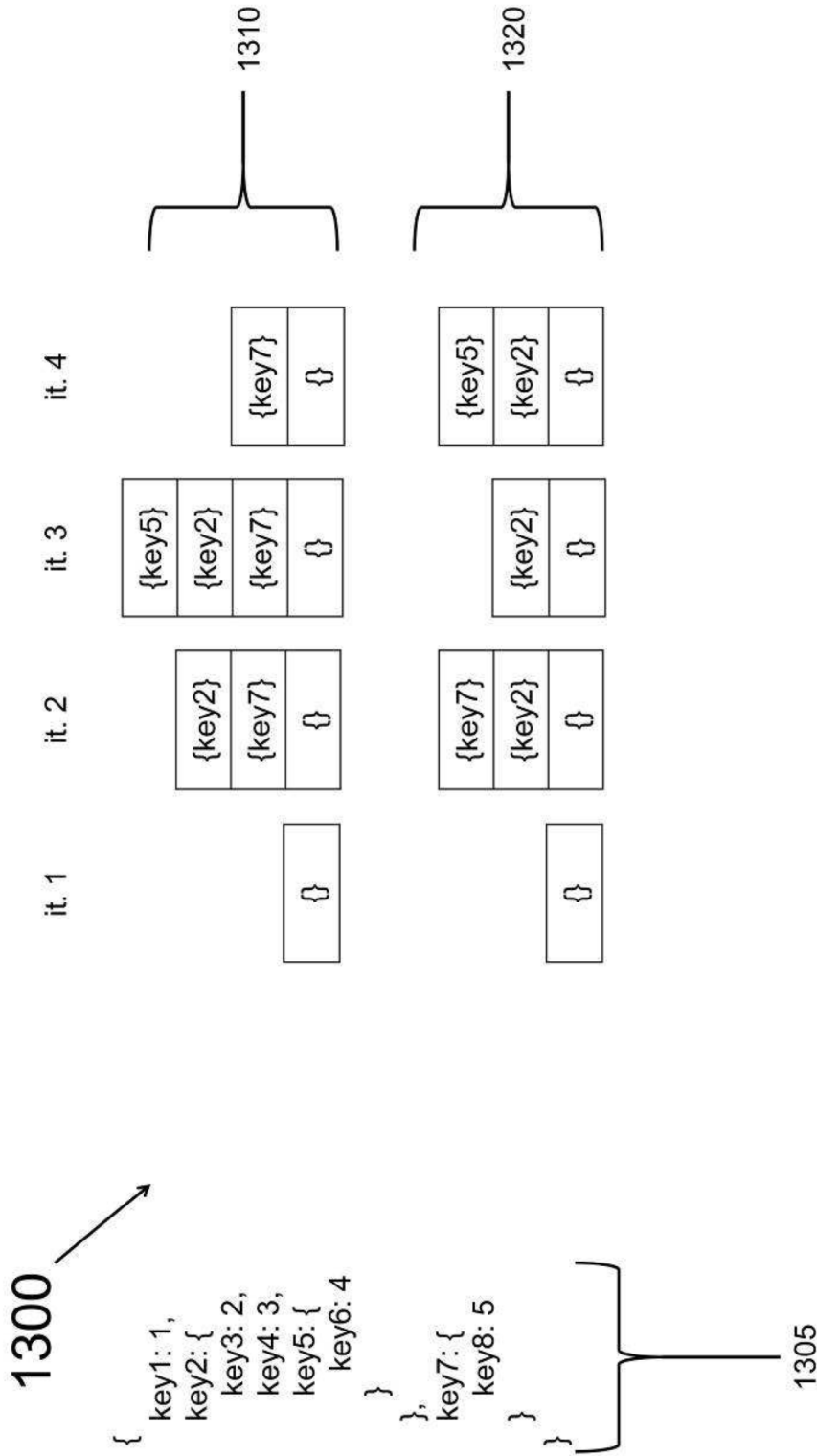


FIG. 13



②① N.º solicitud: 201631481

②② Fecha de presentación de la solicitud: 18.11.2016

③② Fecha de prioridad:

INFORME SOBRE EL ESTADO DE LA TECNICA

⑤① Int. Cl.: **G06F17/30** (2006.01)

DOCUMENTOS RELEVANTES

Categoría	⑤⑥ Documentos citados	Reivindicaciones afectadas
X	CA 2860322 A1 (AMIATO INC) 27/06/2013, Descripción: párs. 68-77, 144-149	1-16
A	US 9418085 B1 (SHIH KATHRYN MARIE et al) 16/08/2016. Todo el documento.	1
A	US 2016246861 A1 (MERRIMAN DWIGHT et al.) 25/08/2016. Todo el documento.	1
A	US 2016275201 A1 (LI HAIFENG et al.) 22/09/2016, Todo el documento.	1
A	US 2012246190 A1 (SURTANI MANIK et al.) 27/09/2012. Todo el documento.	1

Categoría de los documentos citados

X: de particular relevancia

Y: de particular relevancia combinado con otro/s de la misma categoría

A: refleja el estado de la técnica

O: referido a divulgación no escrita

P: publicado entre la fecha de prioridad y la de presentación de la solicitud

E: documento anterior, pero publicado después de la fecha de presentación de la solicitud

El presente informe ha sido realizado

para todas las reivindicaciones

para las reivindicaciones nº:

Fecha de realización del informe
05.02.2018

Examinador
M. Muñoz Sanchez

Página
1/4

Documentación mínima buscada (sistema de clasificación seguido de los símbolos de clasificación)

G06F

Bases de datos electrónicas consultadas durante la búsqueda (nombre de la base de datos y, si es posible, términos de búsqueda utilizados)

INVENES, EPODOC, WPI

Fecha de Realización de la Opinión Escrita: 05.02.2018

Declaración

Novedad (Art. 6.1 LP 11/1986)	Reivindicaciones 14	SI
	Reivindicaciones 1-13, 15-16	NO
Actividad inventiva (Art. 8.1 LP11/1986)	Reivindicaciones	SI
	Reivindicaciones 14	NO

Se considera que la solicitud cumple con el requisito de aplicación industrial. Este requisito fue evaluado durante la fase de examen formal y técnico de la solicitud (Artículo 31.2 Ley 11/1986).

Base de la Opinión.-

La presente opinión se ha realizado sobre la base de la solicitud de patente tal y como se publica.

1. Documentos considerados.-

A continuación se relacionan los documentos pertenecientes al estado de la técnica tomados en consideración para la realización de esta opinión.

Documento	Número Publicación o Identificación	Fecha Publicación
D01	CA 2860322 A1 (AMIATO INC)	27.06.2013
D02	US 9418085 B1 (SHIH KATHRYN MARIE et al.)	16.08.2016
D03	US 2016246861 A1 (MERRIMAN DWIGHT et al.)	25.08.2016
D04	US 2016275201 A1 (LI HAIFENG et al.)	22.09.2016
D05	US 2012246190 A1 (SURTANI MANIK et al.)	27.09.2012

2. Declaración motivada según los artículos 29.6 y 29.7 del Reglamento de ejecución de la Ley 11/1986, de 20 de marzo, de Patentes sobre la novedad y la actividad inventiva; citas y explicaciones en apoyo de esta declaración

Se considera D01 el documento más próximo del estado de la técnica al objeto de la solicitud.

Reivindicaciones independientes

Reivindicación 1: El documento D01 divulga un método de mapeo (conversión) de un conjunto de pares clave-valor (definidos en un documento) en estructuras tabulares de filas y columnas, procesando las instancias de los valores que sean compuestos de forma recursiva para crear subestructuras tabulares a partir de la clave del par clave-valor. El documento puede ser de tipo XML o JSON. La estructura tabular, puede ser una base de datos relacional. Implícitamente la estructura de datos temporal utilizada para el procesamiento recursivo es una pila (Descripción: párs. 68-77, 144-149).

Por tanto el documento D01 afecta a la novedad de la reivindicación 1 según el art. 6.1 de la Ley 11/86 de Patentes.

Reivindicación 16: el sistema reivindicado se corresponde directamente con el método de la reivindicación 1 y, por tanto, el documento D01 también afecta a la novedad de la reivindicación 16 según el art. 6.1 de la Ley 11/86 de Patentes.

Reivindicaciones dependientes

Reivindicaciones 2-13, 15: El contenido de estas reivindicaciones aparece también documento D01. Así, el documento D01 también afecta a la novedad de las reivindicaciones 2-13, 15 según el art. 6.1 de la Ley 11/86 de Patentes.

Reivindicación 14: El uso de una lista enlazada en lugar de una pila como estructura de almacenamiento es una alternativa comúnmente conocida y, por tanto, evidente para el experto en la materia. En consecuencia, el documento D01 afecta a la actividad inventiva de la reivindicación 14 según el art. 8.1 de la Ley 11/86 de Patentes.