



- (51) **International Patent Classification:**
G06F 17/00 (2006.01) *G06F 17/30* (2006.01)
- (21) **International Application Number:**
PCT/US2014/029046
- (22) **International Filing Date:**
14 March 2014 (14.03.2014)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (71) **Applicant:** HEWLETT-PACKARD DEVELOPMENT COMPANY, LP [US/US]; 11445 Compaq Center Drive W, Houston, TX 77070 (US).
- (72) **Inventors:** VARADARAJAN, Ramakrishna R.; 150 Cambridge Park Drive, Cambridge, MA 02140 (US). FINNERTY, James L.; 150 Cambridge Park Drive, Cambridge, MA 02140 (US).
- (74) **Agents:** SAGALEWICZ, Angela M. et al.; Hewlett-packard Company, Intellectual Property Administration, 3404 E. Harmony Road, Mail Stop 35, Fort Collins, CO 80528-9599 (US).
- (81) **Designated States** (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,

BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) **Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to the identity of the inventor (Rule 4.17(i))*

Published:

— *with international search report (Art. 21(3))*

(54) **Title:** COLUMN STORE DATABASE COMPRESSION

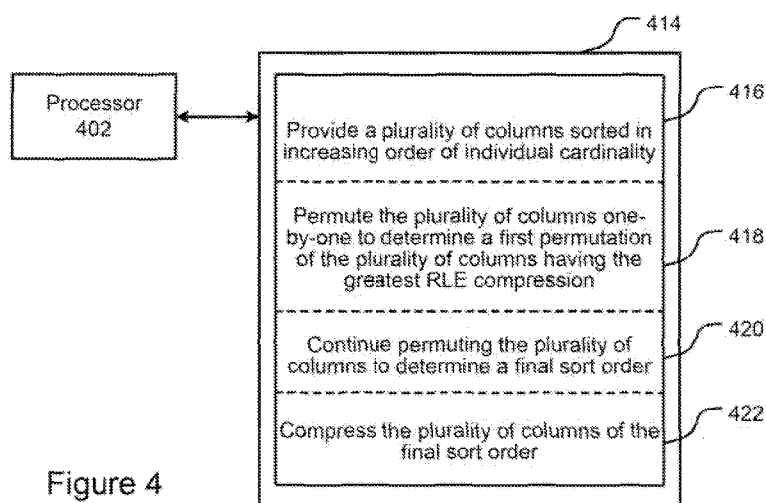


Figure 4

(57) **Abstract:** Described are methods for data compression of a column store database. A method may include providing a plurality of columns sorted from a first position to a last position in increasing order of individual cardinality, permuting columns of the plurality of columns one-by-one to a second position of the plurality of columns, except for the column at the first position to determine a first permutation of the plurality of columns having the greatest run-length encoding (RLE) compression, and permuting columns of the first permutation one-by-one to a third position, except for columns at the second position and the first position, to determine a second permutation having the greatest RLE compression. The method may further include continuing permuting the plurality of columns to determine a final sort order and compressing columns of the final sort order using RLE compression.



COLUMN STORE DATABASE COMPRESSION

Background

[0001] Databases are organized collections of data that can include a collection of records, each record having data pertaining to multiple fields or parameters. Some databases may be represented as a table in which the rows correspond to records and the columns correspond to fields. The intersection of a record (row) and field (column) is termed a "cell" and typically stores the value for a field parameter for a particular database record. Other database types, e.g., relational, hierarchical, and network databases, can have multiple related tables, each with records, fields, and cells.

[0002] While some databases may have only a few cells, others may have over a billion. The amount of data contained in databases may vary significantly. To reduce the amount of physical storage required for database, databases can be compressed.

Brief Description of the Drawings

[0003] The detailed description section references the drawings, wherein:

Figure 1 is a block diagram of an example system endowed with a database manager to compress a column store database of the system;

Figure 2 is a flowchart of an example method for compressing data in a column store database;

Figure 3 is a flowchart of another example method for compressing data in a column store database; and

Figure 4 is a block diagram showing an example tangible, non-transitory, machine-readable medium that stores code adapted to compress data in a column store database;

all in which various embodiments may be implemented.

[0004] Examples are shown in the drawings and described in detail below. The drawings are not necessarily to scale, and various features and views of the drawings may be shown exaggerated in scale or in schematic for clarity and/or conciseness. The same part numbers may designate the same or similar parts throughout the drawings.

Detailed Description of Embodiments

[0005] In a column-organized database (a "column store"), tabular data may be organized into projections that have a specific sort order, and data may be physically clustered by column. As a result of the sort order, non-unique columns appearing early in the sort order may have an opportunity for run-length encoding. In some cases, the columns may include a number of correlated pairs or sets of columns, which may also provide an opportunity for run-length encoding to provide even further data compression.

[0006] Described herein are various implementations of methods, systems, and computer-readable media for data compression of a column store database. A method may include permuting the columns within a sorted projection to exploit correlations among the columns, and thereby to achieve greater run-length encoding (RLE) compression. In some implementations, the method may include sorting a plurality of columns from a first position to a last position in increasing order of individual cardinality, permuting columns of the plurality of columns one-by-one to a second position of the plurality of

columns, except for columns at the first position, to determine a first permutation of the plurality of columns having the greatest RLE compression, and permuting columns of the first permutation one-by-one to a third position, except for columns at the second position and any preceding position, to determine a second permutation having the greatest RLE compression. The method may further include continuing permuting the plurality of columns to determine a final sort order, and compressing columns of the final sort order using RLE compression.

[0007] Referring now to the drawings, Figure 1 is a block diagram of an example system 100 including a processor 102 and a storage device 104 to store a database 106 comprising a plurality of columns of data. The system 100 further includes a database manager 108 to manage the database 106. The database manager 108 may include permutor 110 and a compressor 112. In various implementations, the storage device 104 may include the database manager 108. In various implementations, the system 100 may be implemented as one or more computing devices. The storage device 104 may comprise a magnetic medium, like one or more hard disk drives.

[0008] In operation, the database manager 108 may be executable by the processor 102 to implement a method for data compression of the database 106. In various implementations, the permutor 110 may permute columns of the database 106 one-by-one into a final sort order, in accordance with the various implementations described herein, and the compressor 112 may compress the columns of the final sort order using RLE compression. For example, in some implementations, the permutor 110 may permute columns of the plurality of columns one-by-one to a second position of the plurality of columns, except for the column at the first position, to determine a first permutation of the plurality of columns having an RLE compression greater than an RLE compression of any other permutation. The permute 110 may continue, for example, with permuting columns of the first permutation one-by-one to a third position of the plurality of columns, except for columns at the second position and any preceding position, to determine a second

permutation of the plurality of columns having an RLE compression greater than an RLE compression of any other permutation. In various implementations, a sorter 109 may sort the plurality of columns from the first position to a last position in increasing order of individual cardinality. In some implementations, an identifier 113 may identify correlated column pairs from the plurality of columns of the database 106 and store in memory, such as, for example, the storage device 104, correlated pairs having correlation strength values greater than a predetermined value. In these latter implementations, the stored correlated pairs may be referenced later by the database manager 108 or other component of the system 100 to facilitate looking up data, in response to a query, for example.

[0009] Figures 2 and 3 are flowcharts of example methods 200, 300, respectively, for compressing data in a column store database, in accordance with various implementations. It should be noted that various operations discussed and/or illustrated may be generally referred to as multiple discrete operations in turn to help in understanding various implementations. The order of description should not be construed to imply that these operations are order dependent, unless explicitly stated. Moreover, some implementations may include more or fewer operations than may be described.

[0010] As shown in Figure 2, the method 200 may begin or proceed with providing a plurality of columns sorted from a first position, $l=1$, to a last position in increasing order of individual cardinality at block 216.

[0011] The method 200 may proceed to block 218 with permuting columns of the plurality of columns one-by-one to a second position of the plurality of columns, except for the column at the first position, to determine a first permutation of the plurality of columns having an RLE compression greater than an RLE compression of any other permutation, whereby RLE compression is a factor of grouping cardinalities at each position, the column data types, column width, or correlations, or a combination thereof. The

method 200 may continue to block 220 with continuing permuting the plurality of columns to determine a final sort order.

[0012] The method 200 may proceed to block 222 with compressing the plurality of columns of the final sort order.

[0013] Turning now to Figure 3, the method 300 may begin or proceed with identifying a plurality of correlated pairs a column store database at block 322. In various implementations, correlated pairs of columns may be identified using a "correlation detection via sampling" (CORDS) technique ("CORDS: Automatic Discovery of Correlation and Soft Functional Dependencies" by Ihab F. Ilyas et al.) or another suitable technique.

[0014] The method 300 may proceed with determining the correlation strength value of the correlated pairs by estimating a grouping cardinality of each pair of the correlated pairs at block 324 and determining, for each of the correlated pairs, the correlation strength value based at least in part on a cardinality of each column of the correlated pair and the estimated grouping cardinality of the correlated pair at block 326. As used herein, "grouping cardinality" may refer to the number of distinct column pair values for a correlated pair as grouped, rather than the number of distinct values of the pair as paired independent, individual columns. In various implementations, estimating the grouping cardinality of each of the correlated pairs may be performed using a probabilistic counting algorithm or another suitable algorithm. The correlation strength for each of the correlated pairs may be based, in various implementations, on the number of distinct values for the pair as independent, non-correlated paired columns and as grouped, correlated paired columns. For example, in various implementations, determining the correlation strength values may include determining the lower-bound (LV) for grouping cardinality (assuming the pairs are correlated), the upper-bound (HV) for grouping cardinality (assuming the pairs are independent), and the actual grouping (V) cardinality (the actual cardinality). In these implementations, the correlation strength values may be calculated

as (HV-V)/(HV-LV). In various implementations, operations 324 and 326 may be limited to correlated pairs having a correlation greater than some predetermined threshold such that only the most correlated pairs are further analyzed. In other implementations, all correlated pairs may be analyzed by operations 324 and/or 326.

[0015] The method 300 may proceed with storing in memory correlated pairs having a correlation strength value greater than a predetermined value at block 328. In various implementations, the stored correlated pairs may be referenced later by the database manager or other component of the system to facilitate looking up data, in response to a query, for example. In other implementations, the operation of block 328 may be omitted altogether.

[0016] The method 300 may proceed with sorting the plurality of columns from a first position, $l=1$, to a last position in increasing order of individual cardinality at block 330.

[0017] The method 200 may proceed to block 332 by permuting columns of the plurality of columns one-by-one to a second position of the plurality of columns, except for the column at the first position, to determine a first permutation of the plurality of columns having an RLE compression greater than an RLE compression of any other permutation, whereby RLE compression is a factor of grouping cardinalities at each position, the column data types, column width, or correlations, or a combination thereof. In this operation, the first permutation may be determined considering the first column against all remaining columns to find the best match for the second position (i.e., the column that when placed at the second position gives the highest RLE compression of the plurality of columns. In other words, at position l in the sort order, all other columns may be moved one-by-one (except any columns before position l , which may remain intact) and each resultant sort order may be evaluated for RLE compression.

[0018] The method 300 may continue permuting the plurality of columns at block 334. For example, after determining the first permutation, the columns of the first permutation may be permuted one-by-one to a third position of the plurality of columns ($l=l+1$, i.e., for the next position), except for columns at the second position and any preceding position, to determine a second permutation of the plurality of columns having an RLE compression greater than an RLE compression of any other permutation, and so on. Permuting may continue until reaching a column having an average run-length less than a predetermined run-length threshold at block 336. In various implementations, the operation of block 334 may be performed as it may be desirable to only perform run-length compression on the best candidates having some minimum run length. For example, in some implementations, an RLE threshold may be either $10/N$ (for a segmented database; N =number of nodes) or 10 (for an unsegmented database). In many implementations, permutations at blocks 332/334/336 may operate like a greedy algorithm such that the next column is compared against only the remaining columns, without backward comparison against columns that have already been determined.

[0019] If the next column has an average run-length less than a predetermined less than the predetermined run-length threshold at block 336, the method 300 may proceed to block 338 with compressing the plurality of columns of the final sort order using RLE compression. In various implementations, one or more of the remaining columns (i.e., columns not included in the final sort order) may be compressed using any suitable method or may remain uncompressed.

[0020] Figure 4 is a block diagram showing an example non-transitory computer-readable storage medium 414 that stores computer-implemented instructions adapted to implement data compression of the database 106, in accordance with the various methods described herein. The machine-readable medium 414 may correspond to any typical storage device that stores computer-implemented instructions, such as programming code, or the like, that may be executed by the processor 402. The computer-readable

media 414 may be or may comprise volatile and/or non-volatile media, such as magnetic media, semiconductor media, and the like.

[0021] When read and executed by the processor 402, the instructions stored on the machine-readable medium 414 are adapted to cause the processor 402 to process instructions 416, 418, 420, and 422. A sorter (such as, e.g., the sorter 109 described herein with reference to Figure 1) may provide a plurality of columns sorted in increasing order of individual cardinality (416). A permutor (such as, e.g., the permutor 110 described herein with reference to Figure 1) may permute the plurality of columns one-by-one to determine a first permutation of the plurality of columns having the greatest RLE compression (418) and continue permuting the plurality of columns until reaching a column having an average run-length less than a predetermined threshold to determine a final sort order (420). A compressor (such as, e.g., the compressor 112 described herein with reference to Figure 1) may compress the columns of the final sort order (420).

[0022] Although certain implementations have been illustrated and described herein, it will be appreciated by those of ordinary skill in the art that a wide variety of alternate and/or equivalent implementations calculated to achieve the same purposes may be substituted for the implementations shown and described without departing from the scope of this disclosure. Those with skill in the art will readily appreciate that implementations may be implemented in a wide variety of ways. This application is intended to cover any adaptations or variations of the implementations discussed herein. It is manifestly intended, therefore, that implementations be limited only by the claims and the equivalents thereof.

CLAIMS

1. A method comprising:
 - sorting a plurality of columns from a first position to a last position in increasing order of individual cardinality;
 - permuting columns of the plurality of columns one-by-one to a second position of the plurality of columns, except for the column at the first position, to determine a first permutation of the plurality of columns having a run-length encoding (RLE) compression greater than an RLE compression of any other permutation;
 - permuting columns of the first permutation one-by-one to a third position of the plurality of columns, except for columns at the second position and the first position, to determine a second permutation of the plurality of columns having an RLE compression greater than an RLE compression of any other permutation;
 - continuing permuting the plurality of columns to determine a final sort order; and
 - compressing the plurality of columns of the final sort order using RLE compression.
2. The method of claim 1, wherein said continuing permuting is performed until reaching a column having an average run-length less than a predetermined run-length threshold.
3. The method of claim 1, wherein said permuting the plurality of columns one-by-one to the second position comprises permuting the plurality of columns based at least in part on data type, column width, correlation, or cardinality, or a combination thereof.

4. The method of claim 3, wherein said permuting the plurality of columns of the first permutation one-by-one to the third position comprises permuting the plurality of columns of the first permutation based at least in part on data type, column width, correlation, or cardinality, or a combination thereof.

5. The method of claim 1, further comprising, prior to said sorting the plurality of columns, identifying a plurality of correlated pairs of the plurality of columns.

6. The method of claim 5, further comprising storing in memory correlated pairs having a correlation strength value greater than a predetermined value.

7. The method of claim 6, further comprising determining the correlation strength values of the correlated pairs by:

- estimating a grouping cardinality of each pair of the plurality of correlated pairs; and

- determining, for each of the correlated pairs, the correlation strength value based at least in part on a cardinality of each column of the correlated pair and the estimated grouping cardinality of the correlated pair.

8. The method of claim 7, wherein said estimating the grouping cardinality of each pair of the correlated pairs comprises using a probabilistic counting algorithm.

9. A system comprising:

- a processor;

- a storage device to store a database comprising a plurality of columns of data; and

- a database manager to manage the database and executable by the processor to:

- provide a plurality of columns sorted in increasing order of individual cardinality;

permute columns of the plurality of columns one-by-one to a second position of the plurality of columns, except for the column at a first position, to determine a first permutation of the plurality of columns having the greatest run-length encoding (RLE) compression;

permute columns of the plurality of columns of the first permutation one-by-one to a third position, except for columns at the second position and the first position, to determine a second permutation of the plurality of columns having the greatest RLE compression; and

compress columns at the third position and preceding positions of the second permutation using RLE compression.

10. The system of claim 9, wherein the database manager is executable by the processor to permute columns of the plurality of columns one-by-one to the second position based at least in part on data type, column width, correlation, or cardinality, or a combination thereof.

11. The system of claim 9, wherein the database manager is further executable by the processor to:

identify a plurality of correlated pairs of the plurality of columns;

estimate a grouping cardinality of each pair of the plurality of correlated pairs; and

determine, for each of the correlated pairs, the correlation strength value based at least in part on a cardinality of each column of the correlated pair and the estimated grouping cardinality of the correlated pair.

12. The system of claim 11, wherein the database manager is further executable by the processor to store in memory correlated pairs having a correlation strength value greater than a predetermined value.

13. A non-transitory computer-readable storage medium storing instructions that, when executed by a processor, cause the processor to:

permute columns of a plurality of columns sorted in increasing order of individual cardinality one-by-one to a second position of the plurality of columns, except for the column at a first position of the plurality of columns, to determine a first permutation of the plurality of columns having the greatest RLE compression;

permute columns of the plurality of columns of the first permutation one-by-one to a third position, except for columns at the second position and the first position, to determine a second permutation of the plurality of columns having the greatest RLE compression;

continue permuting the plurality of columns to determine a final sort order; and

compress the plurality of columns of the final sort order using RLE compression.

14. The non-transitory computer-readable storage medium of claim 14, wherein the instructions, when executed by the processors, further cause the processor to:

identify a plurality of correlated pairs of the plurality of columns;

estimate a grouping cardinality of each pair of the plurality of correlated pairs; and

determine, for each pair of the correlated pairs, the correlation strength value based at least in part on a cardinality of each column of the correlated pair and the estimated grouping cardinality of the correlated pair; and

store in memory correlated pairs having a correlation strength value greater than a predetermined value.

15. The non-transitory computer-readable storage medium of claim 14, wherein said continue permuting is performed until reaching a column having an average run-length less than a predetermined run-length threshold.

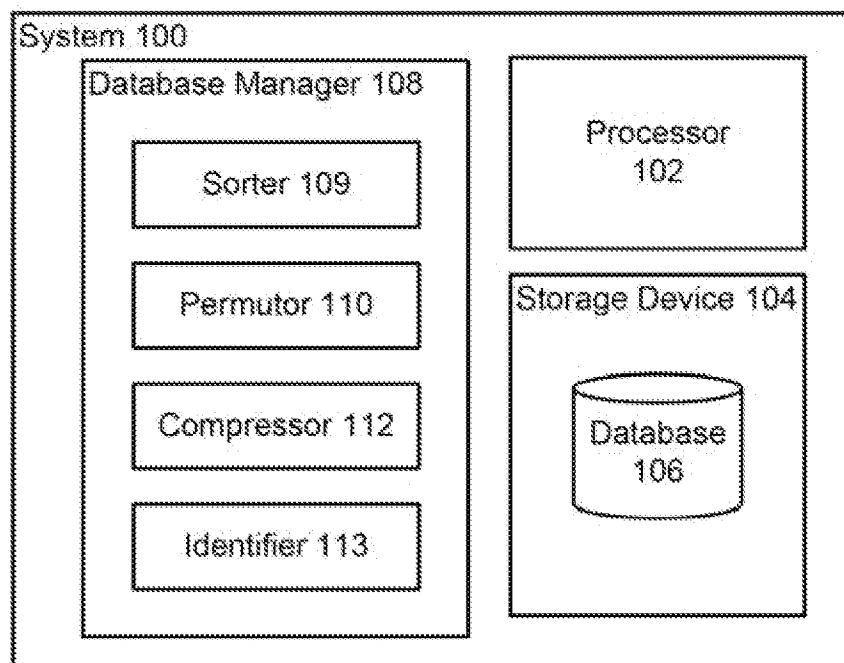


Figure 1

2/4

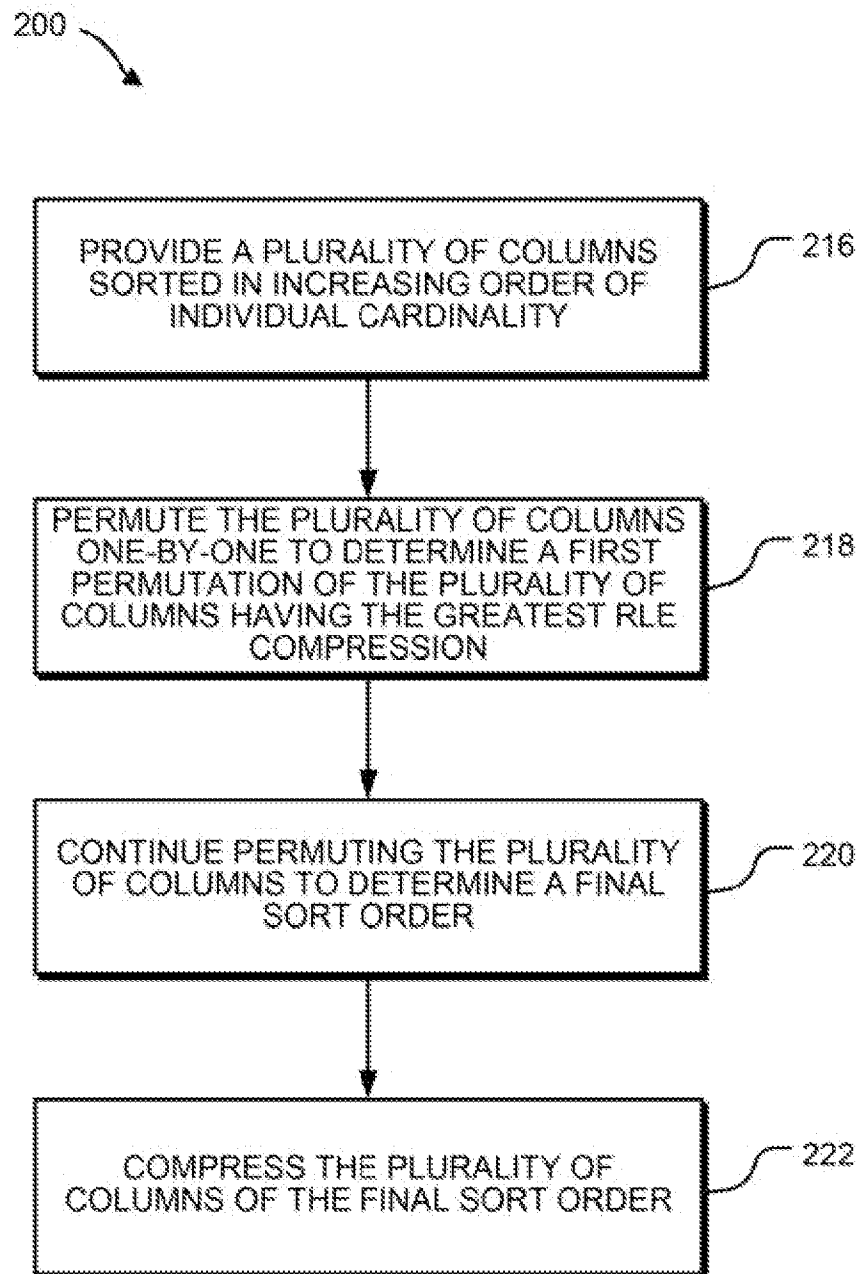
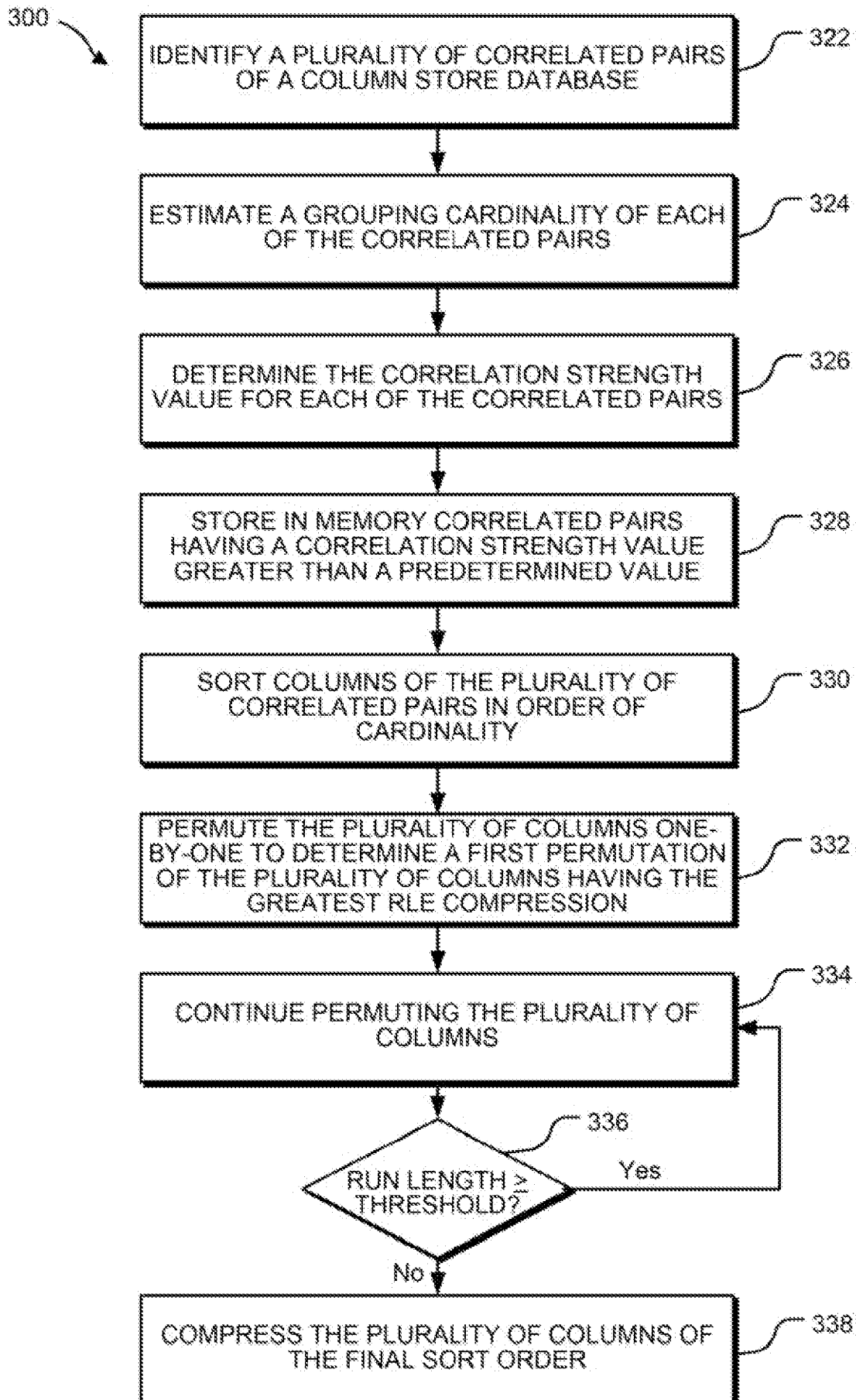


Figure 2

3/4



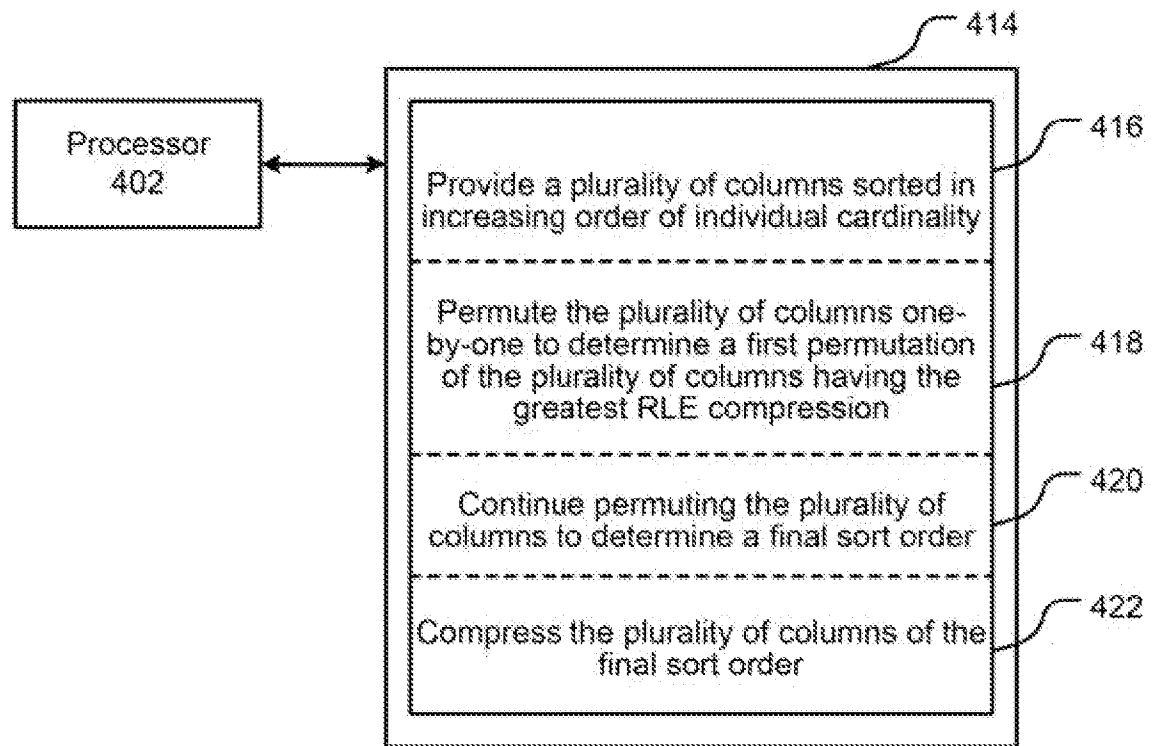


Figure 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2014/029046**A. CLASSIFICATION OF SUBJECT MATTER****G06F 17/00(2006.01)i, G06F 17/30(2006.01)i**

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 17/00; G06F 17/30; G06F 7/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords: database compression, run length encoding, column, sort, permute, and similar terms

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2012-0150877 A1 (RAVISHANKAR RAMAMURTHY et al.) 14 June 2012 See paragraphs [0006], [0036], [0060], and [0066]; claims 1 and 3; and figure 2.	1-15
A	US 2008-0040348 A1 (SHILPA LAWANDE et al.) 14 February 2008 See paragraphs [0043]-[0045] and [0048]; and figure 3.	1-15
A	US 2008-0021914 A1 (ERIC JOHN DAVIES et al.) 24 January 2008 See paragraphs [0025] and [0139]-[0141]; claims 2 and 4; and figure 5.	1-15
A	US 2005-0192994 A1 (DONALD F. CALDWELL et al.) 01 September 2005 See paragraphs [0006], [0016]-[0025], and [0046]-[0049]; and figure 1A.	1-15
A	US 2011-0213766 A1 (MINGSHENG HONG et al.) 01 September 2011 See paragraphs [0004], [0027]-[0032], and [0116]; claim 1; and figure 1A.	1-15



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

08 December 2014 (08.12.2014)

Date of mailing of the international search report

08 December 2014 (08.12.2014)

Name and mailing address of the ISA/KR

International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701,
Republic of Korea

Facsimile No. +82-42-472-7140

Authorized officer

NHO, Ji Myong

Telephone No. +82-42-481-8528



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2014/029046

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012-0150877 A1	14/06/2012	None	
US 2008-0040348 A1	14/02/2008	EP 2052316 A2 EP 2052316 A4 EP 2587365 A2 EP 2587365 A3 WO 2008-016877 A2 WO 2008-016877 A3	29/04/2009 30/06/2010 01/05/2013 28/08/2013 07/02/2008 23/10/2008
US 2008-0021914 A1	24/01/2008	CA 2645354 A1 CA 2645354 C GB 0900967 D0 GB 2454824 A US 8077059 B2 WO 2008-009135 A2 WO 2008-009135 A3	24/01/2008 11/09/2012 04/03/2009 20/05/2009 13/12/2011 24/01/2008 03/02/2011
US 2005-0192994 A1	01/09/2005	US 6959300 B1 US 7720878 B2	25/10/2005 18/05/2010
US 2011-0213766 A1	01/09/2011	EP 2539831 A1 US 8290931 B2 WO 2011-103580 A1	02/01/2013 16/10/2012 25/08/2011