(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
15 December 2005 (15.12.2005)

PCT

(10) International Publication Number
WO 2005/119500 A2

(51) International Patent Classification⁷: G06F 17/00

(21) International Application Number: PCT/US2005/014981

(22) International Filing Date: 29 April 2005 (29.04.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/856,704 28 May 2004 (28.05.2004) US

(71) Applicant *(for all designated States except US)*: **VENDAVO INC** [US/US]; 1001 Elwell Court, Palo Alto, California 94303 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **ALBANESE, Michael, J.** [US/US]; 420 Alberto Way #29, Los Gatos, California 95032 (US). **GONZALEZ-CALONI, Rafael, A** [US/US]; 2 Breaker Lane, Redwood Shores, California 94065 (US).

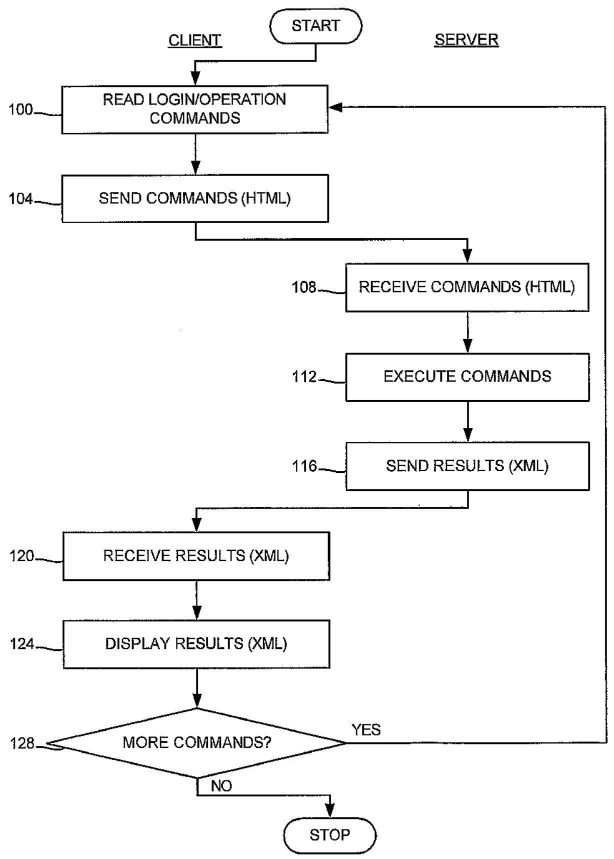
(74) Agent: **KALI, Damon, K, I.**; 10121 Miller Avenue #201, IPSG P.C., Cupertino, California 95014 (US).

(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH,

[Continued on next page]

(54) Title: SYSTEM AND METHOD FOR DISPLAYING PRICE MODELING DATA



(57) **Abstract:** The present invention allows a user make filter, zoom, and roll up selections and apply those selections to a desired price modeling data set. As such the present invention provides a system configured to suitably display price modeling data having a data selection module for selecting at least one subset of the price modeling data, a data aggregation module for aggregating at least one subset of the price modeling data, and a display module configured to display at least one of the at least one selected subset and the at least one aggregated subset of the price modeling data.

GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SYSTEM AND METHOD FOR DISPLAYING PRICE MODELING DATA

BACKGROUND

[0001] Price modeling using a variety of analytical methods has been described in many publications including, for example, economic treatises, textbooks, and patents. Numerical information culled from vast data sets containing numerous transaction based sales operations may be analyzed and displayed in any of a number of different ways such as through word processing, spreadsheets, and graphical software programs. Modeling this numerical information visually in a price modeling context presents various challenges to designers and analysts.

[0002] In order to effectively communicate via visual display methods information gleaned from data sets, designers face many challenges. In one instance the sheer number of data entries in a transaction based data warehouse can typically exceed many millions of transactions. Displaying millions of individual entries on a single display will not generally provide an analyst with useful information. That is, simply graphing a large number of entries alone will not generally provide useful insight into the characteristic and nature of the data set. In this situation, predictive analysis of the data becomes difficult, if not impossible, as a practical matter. Thus, designers must, in some fashion, aggregate, or allow the user to flexibly aggregate the data in order to display any meaningful characterization of a data set.

[0003] Typically, generally known statistical analysis methods may be employed to aggregate data. Averages, medians, modes, and other statistical methods well known in the art may be utilized to aggregate data so that trends and analysis may be affected. However, in some cases, unique methods of data aggregation may be desirable. For example, an average value of price quotes for a particular region may give a sales person a basis upon which to make a real time price quote to a customer. The average value, in this example, represents a statistical locus around which all the price quotes for that region tend to fall. Averaging has an added benefit of reducing a chosen set of data to a single number thus deriving performance gains when analyzing and manipulating the data set. However, as can be appreciated by one skilled in the art, average values cannot generally account for relative comparisons between groups of related items. To make a relative comparison between groups of related items, an index may be calculated.

[0004] Indexes, which are generally known in the art, have been employed in a variety of manners. Stock markets, for example, often use indexes as a gauge of general market condition. Index calculations are typically performed in batch processes. As such, indexes are indicators of past performance only and generally cannot be used to compare real-time data changes. Indexing also tends to aggregate data in ways that make it difficult if not impossible to explore lower level data orders such as individual transactions for example. Further, while indexes generally provide a single value as an indicator, they are not generally visualized in other ways. Thus, an innovative index calculation that may be generated in response to real-time data changes, that allows extraction and manipulation of the underlying data, and that may be visualized in new ways is desirable to achieve a more robust and rich analysis.

[0005] Another complication confronting designers in modeling and displaying large transaction based data sets is that in typical legacy systems, aggregating data often results in the loss of individual transaction information due in part to methodologies selected to enhance performance. As noted above, an average value represents a statistical locus around which all values in the data tend to fall. Also noted above, averaging represents a way of reducing a data set. In many cases, however, a finer level of granularity with respect to the data set may be desirable since, in at least some instances, outliers and dispersions of data tend to become aggregated or averaged away resulting in a distortion of original data set. Thus, for example, a data set may be reduced by averaging based on a given criteria (e.g., average prices for a given item by region). The data set, in this example, may be reduced to a single entry for each item in a given region. As an example, the average price of a widget in the western region may be represented as a single entry. Unfortunately, however, once a data set is reduced, the ability to examine a single transaction or even a group of transactions within the data set is typically severely limited or impossible. Thus, methods to recapture and display single or group transactions of a reduced data set may be desirable.

[0006] In view of the foregoing, systems and method for displaying price modeling data are presented herein.

SUMMARY

[0007] The present invention presents systems and methods for displaying price modeling data. The present invention allows a user make filter, zoom, and roll up selections and apply those selections to a desired price modeling data set.

[0008] One embodiment of the present invention provides a system configured to suitably display price modeling data having a data selection module for selecting at least one subset of the price modeling data, a data aggregation module for aggregating at least one subset of the price modeling data, and a display module configured to display at least one of the at least one selected subset and the at least one aggregated subset of the price modeling data.

[0009] Another embodiment of the present invention presents a method for suitably displaying price modeling data. In the method, price modeling data set selections based on user preferences may be made. Price modeling data may also be used to aggregate on user selected data in accordance with the present invention. The selections and the aggregated selections may be displayed in any combination.

[0010] In still other embodiments of the present invention provides a computer program product in a computer readable media configured to suitably display price modeling data. The compute program product includes a data selection module for selecting at least one subset of the price modeling data; a data aggregation module for aggregating the at least one selected subset of the price modeling data; and a display module configured to display at least one of the at least one selected subset and the at least one aggregated subset of the price modeling data.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] Embodiments of the invention may best be understood by reference to the following description taken in conjunction with the accompanying drawings in which:

FIG. 1 is a flow chart illustrating a simplified client and server communication schema in accordance with an embodiment of the present invention;

FIG. 2 is a flow chart illustrating a method of acquiring data from a data source and populating a transaction table in accordance with an embodiment of the present invention;

FIG. 3 is a flow chart illustrating a method of creating an index in accordance with an embodiment of the present invention;

FIG. 4 is a flow chart illustrating a number of client operations in accordance with an embodiment of the present invention;

FIG. 5 is a flow chart illustrating a method of filtering data in accordance with an embodiment of the present invention;

FIG. 6 is a flow chart illustrating a method of zooming in on data in accordance with an embodiment of the present invention; and

FIG. 7 is a flow chart illustrating a method of rolling up data in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION

[0012] FIG. 1 is a flow chart illustrating a simplified client and server communication system in accordance with an embodiment of the present invention. In general, clients are available via the Internet and may reside locally or remotely from a server or group of servers. Thus, the context in which the foregoing communications are made is illustrated here. At a step 100, client login and operation(s) may be made. Login and operation(s) may be enabled to generate commands that are sent to a server at step 104. Login commands and operation(s) commands may be made by the user through a browser interface which generates HTML data to be forwarded to a server or group of servers. The described invention, however, is not limited to Internet browsers and may be implemented with any suitable network communication protocol. The commands sent to servers are received by the servers at step 108. Commands sent by the client may then be executed at step 112. In this manner, the computational horsepower necessary to manipulate data set commands is centralized. Thus, the client need only contain sufficient computational power to send commands, receive results, and display results. As can be appreciated by one skilled in the art, centralizing computational horsepower lowers cost to entry for client side users. Network and software maintenance may also realize efficiencies due to centralization.

[0013] Once commands are executed by a server or servers, results may be sent to a client in XML at step 116. XML does not return transactional data to be processed and displayed by a client. Rather, as noted above, the command computations are executed at a server level which then returns XML to a client to display computational results. In this manner, results may be displayed by any method capable of displaying XML data. The described invention, however, is not limited to sending and receiving XML data and may thus be implemented using any suitable network communication protocol. Results sent in step 116 may be received by a client at step 120 and displayed by a client at step 124. In the example embodiment provided, any XML capable browser may display results received at step 120.

[0014] Once results are displayed, a client determines whether more commands are to be processed at step 128. If more commands need to be processed, the method returns to step 100 and processes commands as described above. When all commands have been completed, a user may sign off and the method ends.

[0015] FIG. 2 is a flow chart illustrating a method of acquiring data from a data source and populating a transaction table in accordance with an embodiment of the present invention. At a step 200, a client configuration file is read. A client configuration file includes a number of parameters including, for example, mapping parameters and calculation parameters. As can be appreciated, mapping parameters may be generated to map data stored in a data warehouse prior to importing stored data into a desired data structure. In this manner, data from legacy systems may be preserved and formatted for a particular application. In addition, calculation parameters may be generated to create new data structures based on an imported data set. For example, a particular user may be interested in the difference between a statistical mean of a data set and a particular transaction entry as illustrated by the following equation:

$$\text{calculated value} = \text{entry value} - \text{mean value}$$

Equation 1: Example calculated value

[0016] The resulting calculated value may then be stored in a new data structure along with imported data. Any number of user defined calculation parameters may be incorporated into a client configuration file. In one embodiment of the present invention, an index is a calculation parameter. Index generation will be discussed in further detail for FIG. 3 below. In one embodiment of the present invention, a calculated value may be added to a transaction table created in step 224, which will be discussed in further detail below.

[0017] After a client configuration file is read, the method determines whether a transaction table all ready exists at step 204. Transaction tables, at a general level, represent the data extracted from the data source in table format along with calculated values. More particularly, a transaction table is a two dimensional table with rows and columns. Each row, in some embodiments, represent a transaction while each column represents some characteristic, description, or calculated value corresponding to a row. It may be appreciated that the designation of rows and columns is for convenience only since a transaction table may also be organized as columns representing transactions and rows representing characteristics, descriptions, or calculated values corresponding to a column.

[0018] Thus, for example, a transaction table may be populated as follows:

transaction	date	price	region	= Price – Average Price	...
1	4/4/04	4.50	SW	-.775	...
2	4/4/04	7.50	NE	2.243	...
3	4/5/04	3.75	S	-1.525	...
4	4/3/04	5.35	W	.075	...
...	...	...	...	...	...

Table 1: Example Transaction Table

[0019] In the above example, a transaction table is populated with four transactions and four columns of data corresponding to each transaction. Some of the entries may be defined as measurements which are typically numeric and others may be defined as dimensions which are typically descriptive elements of the transaction. It is noted that this table reflects actual values for the given fields; however, in some embodiments the values may be converted into expression objects which have the advantage of being syntactically equivalent so that operations on both measurements and definitions may be similarly processed. It is also noted that the above table contains only four transaction entries and five descriptive fields that are either measurements or dimensions. The present invention contemplates many more transactions and many more columns. In a preferred embodiment, the table may contain up to approximately three million records. In other preferred embodiments, the table may contain up to approximately one million records. Memory limitations described in these embodiments are due in part to the limitations associated with 32-bit architecture currently supported by most IT organizations. However, this limitation may be overcome by porting the present invention to a higher capacity 64-bit architecture and above.

[0020] If a transaction table is found at step 204, the method purges a found transaction table from memory at step 208. In this manner, old transaction tables are removed before new tables are formed thus avoiding data corruption. Once an old transaction table is purged, the method continues to step 212. If a transaction table is not found at step 204, the method continues to step 212 where data is requested from a data source in accordance with a configuration file read at step 200. A data source may be any of a number of data sources well-known in the art including, for example, Oracle databases or SAP databases. As noted above, configuration files map data source data to conform to a desired transaction table format. Once data has been extracted from a data source, extracted data is transformed at step 220 using a configuration file read at step 200. As noted above, a configuration file may contain, along with other parameters, calculation parameters to be applied to extracted data sets. For example, in Table 1 above, a calculated parameter illustrated is defined by the equation: calculated value = price – average price. Thus, for each transaction a value may be transformed by a calculation applied a transaction. In a preferred embodiment, at least one index is calculated. Indexes may be, without limitation, numeric values, or some other relative absolute value. Index calculations are discussed in further detail below for FIG. 3.

[0021] Once extracted data is transformed, a resulting data set is loaded into memory (RAM) as a transaction table containing all desired transactions from a data source along with all transformed data generated at step 224. As can be appreciated by one skilled in the art, 32-bit platforms can only address up to 4.0 gigabytes of physical memory (RAM) (i.e., $2^{32} = 4000$ million). In WINDOWS ™ operating system, processes are limited to 2.0 gigabytes of RAM. In some embodiments using Java Virtual Machine (JVM), the process space is further limited to 1.5 gigabytes of RAM. Thus, in a preferred embodiment, the amount of usable memory (RAM) for a transaction table is approximately 1.5 gigabytes. More preferably, the amount of usable memory (RAM) for a transaction table is approximately 1.0 gigabytes. As noted above, in systems utilizing 64-bit platforms, no such memory limitations are contemplated. After a transaction table is loaded into memory (RAM) at step 224, the system then waits for user input at as step 228 of the type described, for example, in FIG. 4.

[0022] FIG. 3 is a flow chart illustrating a method of creating an index in accordance with an embodiment of the present invention. An index is essentially a summary measure that states a relative comparison between groups of related items. A user may define an index to compare a metric from a selected transaction against a similar metric of a set of definable transactions. For example, a margin index (i.e., index) can be defined to express the relationship between the net margin of a transaction (i.e., selected transaction) and the average margin from the peer customers (i.e., set of definable transactions). Moreover the indexes can be rolled up by any arbitrary grouping such as by customer, by product, etc. Roll up, which is a data aggregation feature, will be discussed in further detail below for FIG. 7. Further, indexes, as contemplated by the present invention, may be used to drive future behavior in that an index may be compared against a current metric as opposed to typical use that uses indices to characterize past performance. Conventionally, a commonly used market indicator, for example, is the Standard & Poor's (S&P) composite index. The S&P composite index includes 500 of the largest (in terms of market value) stocks in the United States. As such, the S&P composite index indicates the general trend of a common-stock portfolio's past performance. However, the S&P composite index is not generally used to drive future behavior. That is, it is not generally used to indicate whether to purchase an individual stock.

[0023] As noted above for FIG. 2 at step 220, data from a data source is transformed in accordance with a configuration file. A configuration file may contain calculation parameters to be applied to an extracted data set. In a preferred embodiment, at least one index is calculated. It can be appreciated that one or many transformation events may occur at step 220. Turning to FIG. 3, FIG. 3 represents at least one transformation event corresponding to step 220. At a step 300, index support is called. A unique index column for the transaction table may be created at a step 304. An index column stores a value of the resulting index. An index base is then calculated at a step 308. An index base represents the denominator of an index. An index base may be a user defined set of transactions like, for example, the average of the list price less the invoice price in the south west region over the last 2 quarters. In this example, the average of the list price less the invoice price term results in a value while the regional and time interval terms are descriptive and serve to delimit an entire data set to a selected data set. Many different index bases are contemplated and may be utilized in the present invention without restriction.

[0024] An index numerator is then read or calculated at a step 312. A numerator may be a data entry like, for example, a list price or it may be a calculated value like, for example, list price less invoice price. In either case, a numerator may be stored in a transaction table. Once an index denominator and an index numerator are calculated at steps 308 and 312 respectively, an index may be calculated at step 316 and loaded into an index column at step 320. Using the above mentioned denominator and numerator, an example of an index formulation is illustrated according to the following equation:

$$Index = \Pr ice \operatorname{Re} alization = \frac{List \Pr ice - Invoice \Pr ice}{\begin{gathered} AVG(List \Pr ice - Invoice \Pr ice) \\ (region = southwest : time = previous2quarters) \end{gathered}}$$

Equation 2: Example index calculation

[0025] The method then determines whether another desired transaction exists for which an index may be calculated at step 324. If another desired transaction exists, the method returns to step 312 and cycles until all desired transactions are processed. When all transactions are processed, the method then determines whether all indexes have been processed at step 328. If more index calculations need processing, the method returns to step 304 and cycles until all index calculations have been processed. When all indexes have been processed, the method either returns to step 224 if all transformations are complete, or continues transforming data. Populating a transaction table to be loaded into memory (RAM) has been described above.

[0026] It can be appreciated that indexes, as disclosed may be calculated in real-time using a current data set. Furthermore, transforming data in accordance with the present invention does not result in the loss of information because a transaction table, upon which an index is calculated, is preserved. Thus, a user may freely explore data that underlies an index calculation, thus yielding a richer research tool. In addition, indexes, as contemplated by the present invention, may be rolled up. Generally speaking, roll up allows a user to summarize by field a set of data. In this example, because an underlying data set for a given index calculation is available; a user may select data descriptors that may further delineate an index calculation. Roll up will be discussed in further detail below for FIG. 7.

[0027] FIG. 4 is a flow chart illustrating a number of user operations in accordance with an embodiment of the present invention. At a step 400, a user may login to a client. As noted for FIG. 1, the commands generated by a client are transmitted to a server. A server may then return results to a client whereupon results may be graphically displayed for a user. Thus, user display may be updated upon each operation or command. Once a user has logged on, a chart may be selected at a step 404. Charts are graphical illustrations of data sets. In a preferred embodiment, price band, zebra, time series, multiple time series, scatter, and waterfall charts may be selected from the user interface. User interfaces contemplated under the present invention will be discussed in further detail below. Other charts are contemplated by the present invention and may include without limitation, bar charts, line charts, or pie charts.

[0028] After a user has selected a particular chart at step 404, a user may select appropriate axis parameters at a step 408. Axis parameters represent a desired data set to be plotted. For example, price indicators (y-axis parameter) may be plotted against temporal indicators (x-axis parameter) to determine the change in pricing over time. In like manner, temporal indicators (y-axis parameter) may be plotted against price indicators (x-axis parameter). Thus, selection of axis parameters may be highly flexible according to user preference. In some embodiments, indexes may be selected as axis parameters. In other embodiments, axis parameters may be selected from drop down menus that contain many possible parameters in accordance with a corresponding configuration file. Axis parameters may be further selected in any manner known in the art without limitation.

[0029] At another step 412, a user may select any of a number of different filters. Filters will be discussed in further detail below for FIG. 5. In general, filters allow a user to display a subset of the original data set based on a desired criterion. Filters applied in this manner retain the order in which they are applied and may be selectively removed from the order depending on user preferences. Filters are displayed in order of selection for the user. A user may also select a zoom at a step 416. Zoom will be discussed in further detail below for FIG. 6. In general, zoom further limits a displayed data set. In one embodiment, zooming is accomplished by graphically selecting, by mouse drag, an area that a user desires to expand. A resulting zoom is enlarged and displayed for a user containing data encompassed by an area selected. In other embodiments, zoom parameters may be manually entered. Zooms may be displayed in order of selection made by a user.

[0030] will be discussed in further detail below for FIG. 7. In general, roll up allows a user to aggregate data according to a selection criterion. Aggregated data may also be summarized in accordance with a desired axis parameter. For example, if price is displayed as an axis parameter and a user desires to roll up prices according to sales region, then a resulting roll up displays average prices for each sales region. The method continues until the method determines that no more operations are desired at step 424 whereupon the method ends.

[0031] It can be appreciated that the operations described under steps 408 through 420 may be selected in any order in accordance with a user's preferences. Furthermore, selections described under steps 412 and 416 may be multiply selected and preserved in any order in accordance with a user's preferences and may be displayed in a selection list. Still further, selections may be individually or multiply added to or removed from an existing selection list. In some alternate embodiments, filter data sets may be retained in and subsequently recovered from cache. In other alternate embodiments, zoom data sets may be retained in and subsequently recovered from cache. Recovering data sets from cache may realize performance gains and memory efficiencies. Finally, in an embodiment of the present invention, a user may select any type of chart under step 404 while preserving selections made previous to a chart selection. That is, in some embodiments, axis parameters, filters, zoom, and roll up may persist across chart selections.

[0032] FIG. 5 is a flow chart illustrating a method of filtering data in accordance with an embodiment of the present invention. In particular, the flow chart further illustrates step 412 of FIG. 4. At a step 500, filter criterion selected by a user is received. As noted above, filter criterion may be selected or input by a user. Typically, filter criterion limits data in accordance with known operands as for example: =, <, >, □, □, AND, OR, etc. Furthermore, filters may be organized by dimension and by measure. Dimension is an attribute of a transaction that can have any one of a known list of values. For example, every transaction has customer and there is a known set of customers. Thus customer is an example of a dimension. A dimension may have a flat list of values or it may have a hierarchical list. Measure is an attribute that has a numeric value. In some embodiments, a value may be an amount of money. Thus, selection by measure is a selection based on a numeric value.

[0033] After filter criterion is received, an expression object based on a received filter criterion is created at a step 504. Creation of an expression object allows for efficient syntactical processes to be achieved that may result in performance advantages. Once an expression object is created for a filter at step 504, a determination is made as to whether a roll up table is required. As noted above, a roll up allows a user to aggregate data according to a selection criterion. Aggregated data may also be summarized in accordance with a desired axis parameter. Roll up requirements must be considered before a filter is applied to a data set because a rolled up data set may respond differently to a selected filter than an original data set. Roll up will be discussed in further detail below for FIG. 7. If a roll up is required, the method calls roll up manager at a step 512. A roll up table is then returned at step 516 and the method continues to step 520. If the method determines at step 508 that a roll up is not required, the method continues to step 520.

[0034] At a step 520, a row in a table (e.g., transaction table, or roll up table) is evaluated according to a filter expression object. A filter column corresponding to a selected filter may then be created in a table to hold a Boolean result of the evaluation of step 520 at step 524. The method then determines whether a row under examination matches a selected filter (i.e., Boolean = true) at a step 528. If a row under examination matches a selected filter, then that row is added to a row set representing a set of data matching a selected filter at a step 532. The method then determines whether more rows need evaluation at step 536. If more rows need evaluating, the method returns to step 520 and cycles until all rows in a table are evaluated. If, at step 528, a row under examination does not match a selected filter (i.e., Boolean = false), then the row under examination is not added to a row set and the method continues at step 536 to determine whether more rows need evaluation. As noted above, if more rows need evaluating, the method returns to step 520 and cycles until all rows in a table are evaluated.

[0035] When all rows have been evaluated, the method continues at a step 540 to determine whether additional filters have been selected. If additional filters have been selected, the method returns to step 500 and cycles and continues until all filters have been evaluated. The method then ends. Note that each filter selection requires roll up evaluation to assure that tahe roll up is properly applied in a sequence of selected operations.

[0036] FIG. 6 is a flow chart illustrating a method of zooming in on data in accordance with an embodiment of the present invention. In particular, the flow chart further illustrates step 416 of FIG. 4. As noted above, in general, zoom further limits a displayed data set. In one embodiment, zooming is accomplished by selecting an area by mouse drag that a user desires to expand. A resulting zoom is enlarged and displayed for a user containing data encompassed by an area selected. In other embodiments, zoom parameters may be manually entered. A zoom criteria results in what may be thought of as a multiple filter. Thus, for a given chart, if a user selected, by mouse drag, an area of interest, the method would return a set of parameters corresponding to an area of interest. For example, if a chart of price (y-axis) over time (x-axis) is displayed, a user may select an area of a displayed chart by mouse drag. The method would then return upper and lower parameters for both axes corresponding to a selected area. Thus, an area greater than selected upper parameters and less than selected lower parameters may be displayed. In this manner, data is filtered by upper and lower parameters therefore allowing processing using the same logic as for a filter selection.

[0037] At a step 600, zoom criteria selected by a user are received. As noted above, zoom criteria may be selected (by mouse drag) or manually input by a user. Typically, zooms may be organized by dimension and by measure. Dimension is an attribute of a transaction that can have one of a known list of values. For example, every transaction has customer and there is a known set of customers. Thus customer is an example of a dimension. A dimension may have a flat list of values or it may have a hierarchical list. Measure is an attribute that has a numeric value. In some embodiments, a value may be an amount of money. Thus, selection by measure is a selection based on a numeric value.

[0038] After zoom criteria are received, an expression object based on received zoom criteria is created at a step 604. As noted above, creation of an expression object allows for efficient syntactical processes to be achieved that may result in performance advantages. Once an expression object is created for a zoom at step 604, a determination is made as to whether a roll up table is required. As noted above, a roll up allows a user to aggregate data according to a selection criterion. Aggregated data may also be summarized in accordance with a desired axis parameter. Roll up requirements must be considered before a zoom is applied to a data set because a resulting roll up data set may respond differently to a selected zoom than an original data set. Roll up will be discussed in further detail below for FIG. 7. If a roll up is required, the method calls roll up manager at a step 612. A roll up table is then returned at step 616 and the method continues to step 620. If the method determines at step 608 that a roll up is not required, the method continues to step 620.

[0039] At a step 620, a row in the table (e.g., transaction table, or roll up table) is evaluated according to a zoom expression object. A zoom column corresponding to a selected zoom is then created in a table to hold a Boolean result of the evaluation of step 620 at step 624. The method then determines whether a row under examination matches a selected zoom (i.e., Boolean = true) at a step 628. If a row under examination matches a selected zoom, that row under examination is added to a row set representing a set of data matching a selected zoom at a step 632. The method then determines whether more rows need evaluation at step 636. If more rows need evaluating, the method returns to step 620 and cycles until all rows in a table are evaluated. If, at step 628, a row under examination does not match a selected zoom (i.e., Boolean = false), then that row under examination is not added to a row set and the method continues at step 636 to determine whether more rows need evaluation. As noted above, if more rows need evaluating, the method returns to step 620 and cycles until all rows in a table are evaluated.

[0040] When all rows have been evaluated, the method continues at a step 640 to determine whether additional zooms have been selected. If additional zooms have been selected, the method returns to step 600 and cycles and continues until all zooms have been evaluated. The method then ends. Note that each zoom selection requires roll up evaluation to assure that a roll up is properly applied in a sequence of selected operations. Note that the use of an expression object allows for substantially identical syntactical processing and may result in more efficient code.

[0041] FIG. 7 is a flow chart illustrating a method of rolling up data in accordance with an embodiment of the present invention. As noted above, a roll up allows a user to aggregate data according to a user selected criterion. Aggregated data may also be summarized in accordance with a desired axis parameter. For example, a chart displaying price points over time may be rolled up to reduce a selected data set. In this example, a user may, for example, select a roll up based on sales region. The price points would then be sorted and summarized by sale region.

[0042] At a step 700, a roll up selection criterion is read. Roll up selection criterion may be selected by menu or by input as desired by a user. The method continues at a step 704 to determine whether a roll up table matching a roll up selection criterion is available in cache. By using cached tables, the method may achieve performance advantages over prior art methodologies. If a roll up table exists in cache, a cached roll up table is read into memory at a step 720 whereupon the method ends. If a roll up table is not available, the method reads a transaction table at a step 708 that was created at step 224, FIG. 2. A transaction table is the basis upon which a roll up table is cloned at step 712. A cloned roll up table contains all transactions matching a user selected roll up criterion read at step 700. Any transactions not matching a user selected roll up criterion are not cloned into a roll up table. A transaction table that is being cloned remains in memory unaltered. The method then aggregates data based on a selection criterion at step 712 so that summarized data may be displayed for a user. The method then ends.

[0043] While this invention has been described in terms of several preferred embodiments, there are alterations, permutations, modifications and various substitute equivalents, which fall within the scope of this invention. For example, the portfolios illustrated in FIG. 5 are illustrative only and may be organized within the approval hierarchy in numerous wars. It should also be noted that there are many alternative ways of implementing the methods and systems of the present invention. It is therefore intended that the following appended claims be interpreted as including all such alterations, permutations, modifications, and various substitute equivalents as fall within the true spirit and scope of the present invention.

CLAIMS

What is claimed is:

1. A system configured to suitably display price modeling data comprising:

a data selection module for selecting at least one subset of the price modeling data;

a data aggregation module for aggregating the at least one selected subset of the price modeling data; and

a display module configured to display at least one of the at least one selected subset and the at least one aggregated subset of the price modeling data.

2. A method for suitably displaying price modeling data comprising:

selecting at least one subset of the price modeling data;

aggregating the at least one selected subset of the price modeling data; and

displaying at least one of the at least one selected subset and the at least one aggregated subset of the price modeling data.

3. A computer program product in a computer readable media configured to suitably display price modeling data, the computer program product comprising:

a data selection module for selecting at least one subset of the price modeling data;

a data aggregation module for aggregating the at least one selected subset of the price modeling data; and

a display module configured to display at least one of the at least one selected subset and the at least one aggregated subset of the price modeling data.

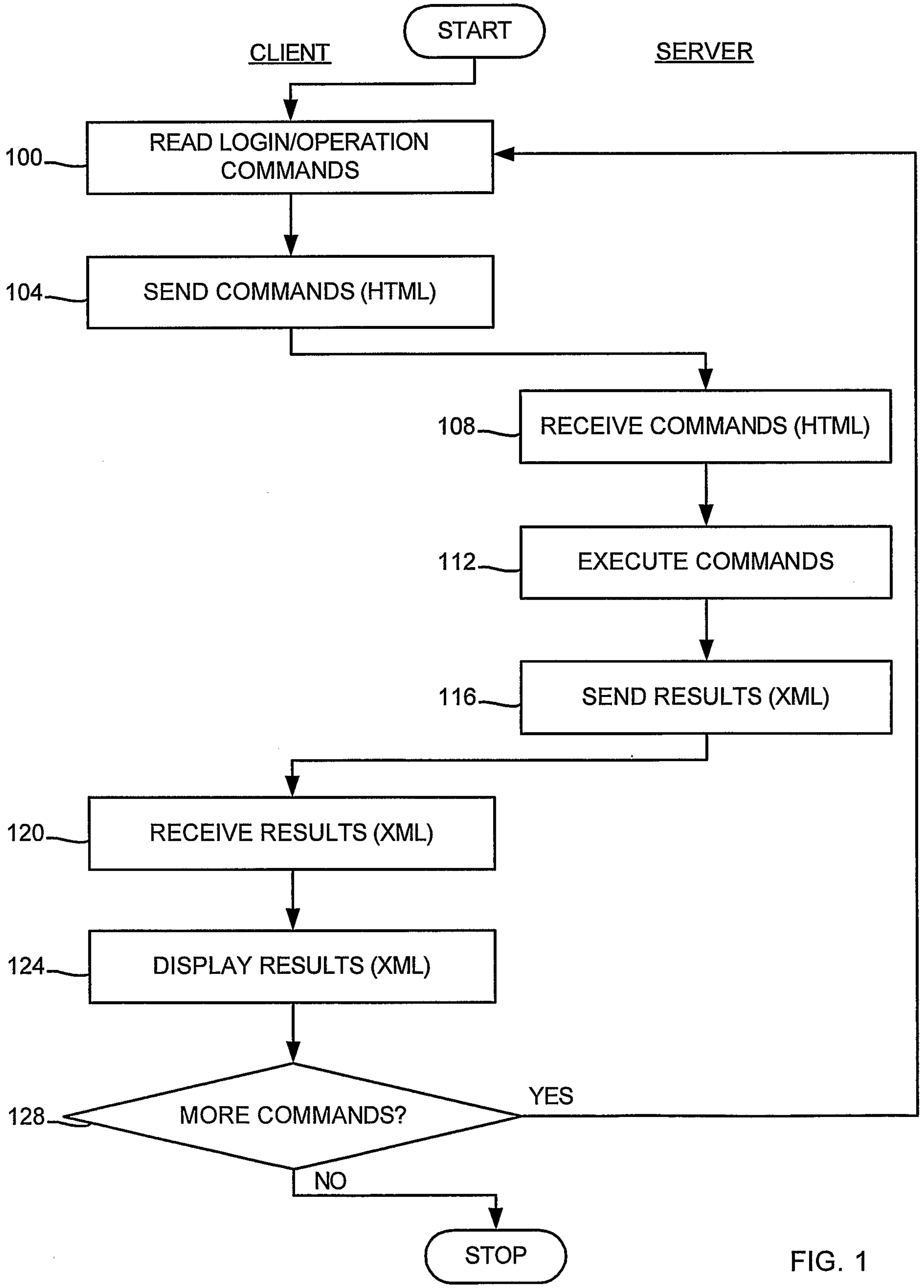

START
CLIENT
SERVER
100
READ LOGIN/OPERATION COMMANDS
104
SEND COMMANDS (HTML)
108
RECEIVE COMMANDS (HTML)
112
EXECUTE COMMANDS
116
SEND RESULTS (XML)
120
RECEIVE RESULTS (XML)
124
DISPLAY RESULTS (XML)
128
MORE COMMANDS?
YES
NO
STOP

FIG. 1

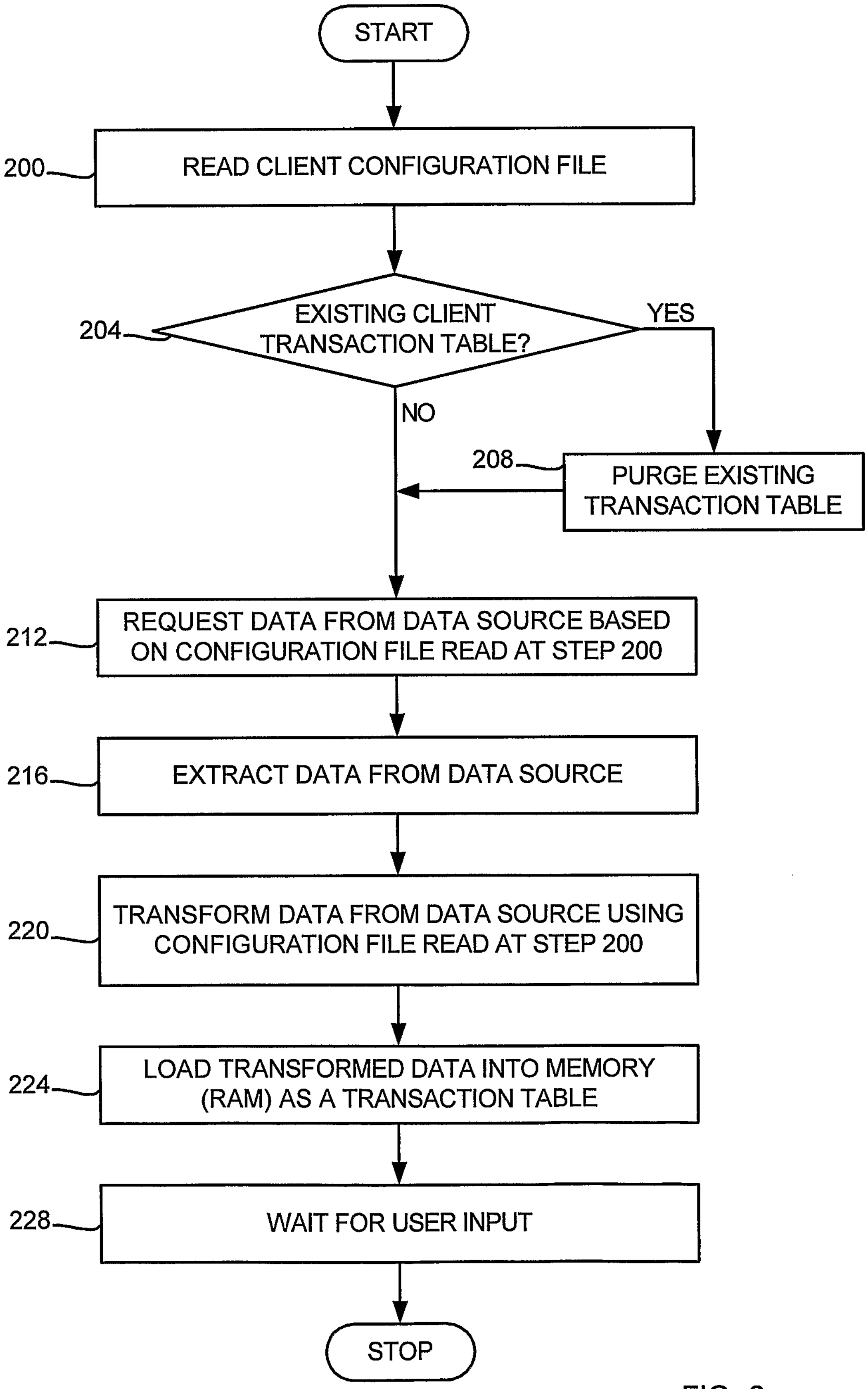

START
200
READ CLIENT CONFIGURATION FILE
204
EXISTING CLIENT TRANSACTION TABLE?
YES
NO
208
PURGE EXISTING TRANSACTION TABLE
212
REQUEST DATA FROM DATA SOURCE BASED ON CONFIGURATION FILE READ AT STEP 200
216
EXTRACT DATA FROM DATA SOURCE
220
TRANSFORM DATA FROM DATA SOURCE USING CONFIGURATION FILE READ AT STEP 200
224
LOAD TRANSFORMED DATA INTO MEMORY (RAM) AS A TRANSACTION TABLE
228
WAIT FOR USER INPUT
STOP

FIG. 2

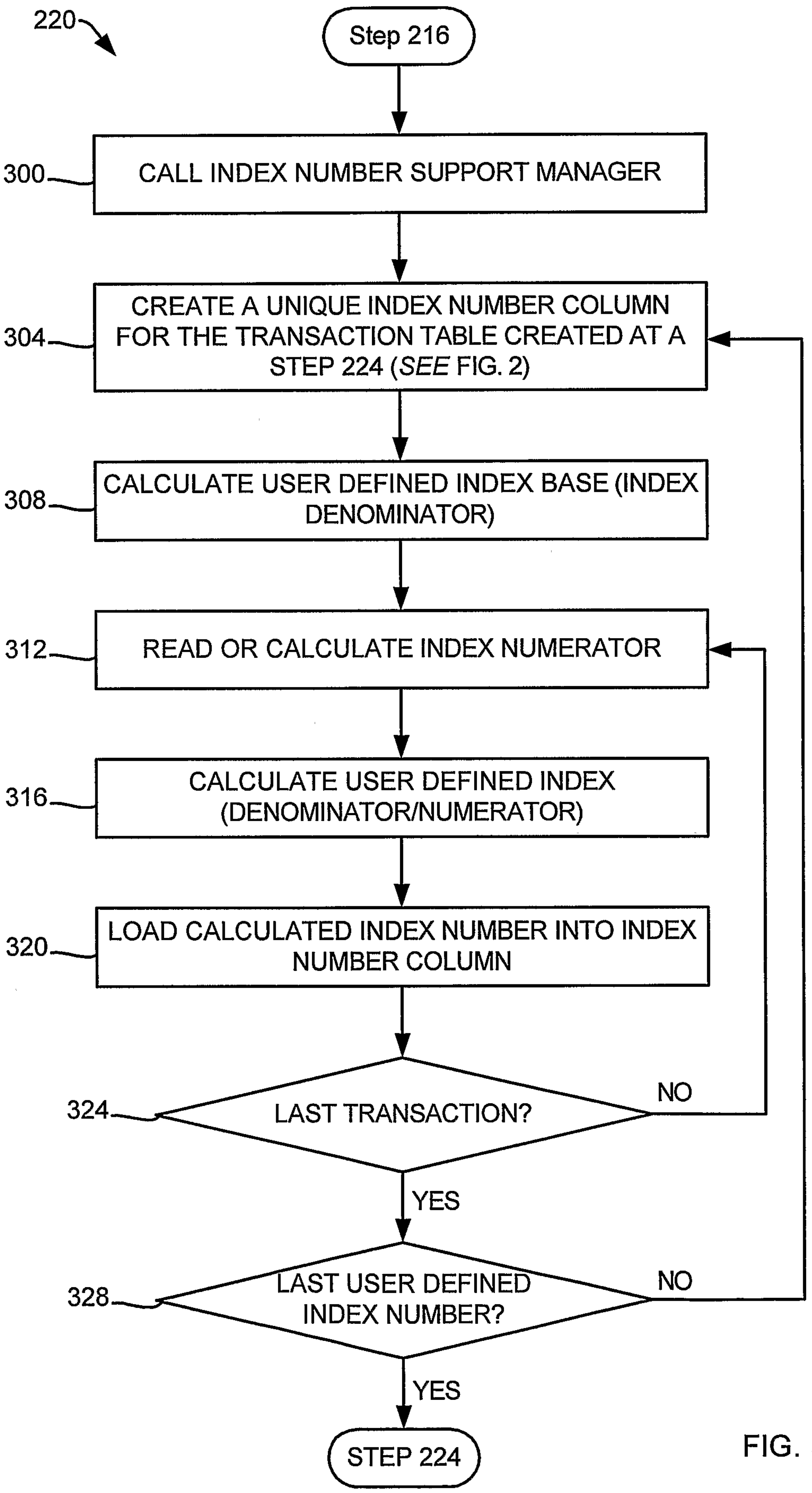

220
Step 216
300
CALL INDEX NUMBER SUPPORT MANAGER
304
CREATE A UNIQUE INDEX NUMBER COLUMN FOR THE TRANSACTION TABLE CREATED AT A STEP 224 (*SEE* FIG. 2)
308
CALCULATE USER DEFINED INDEX BASE (INDEX DENOMINATOR)
312
READ OR CALCULATE INDEX NUMERATOR
316
CALCULATE USER DEFINED INDEX (DENOMINATOR/NUMERATOR)
320
LOAD CALCULATED INDEX NUMBER INTO INDEX NUMBER COLUMN
324
LAST TRANSACTION?
NO
YES
328
LAST USER DEFINED INDEX NUMBER?
NO
YES
STEP 224

FIG. 3

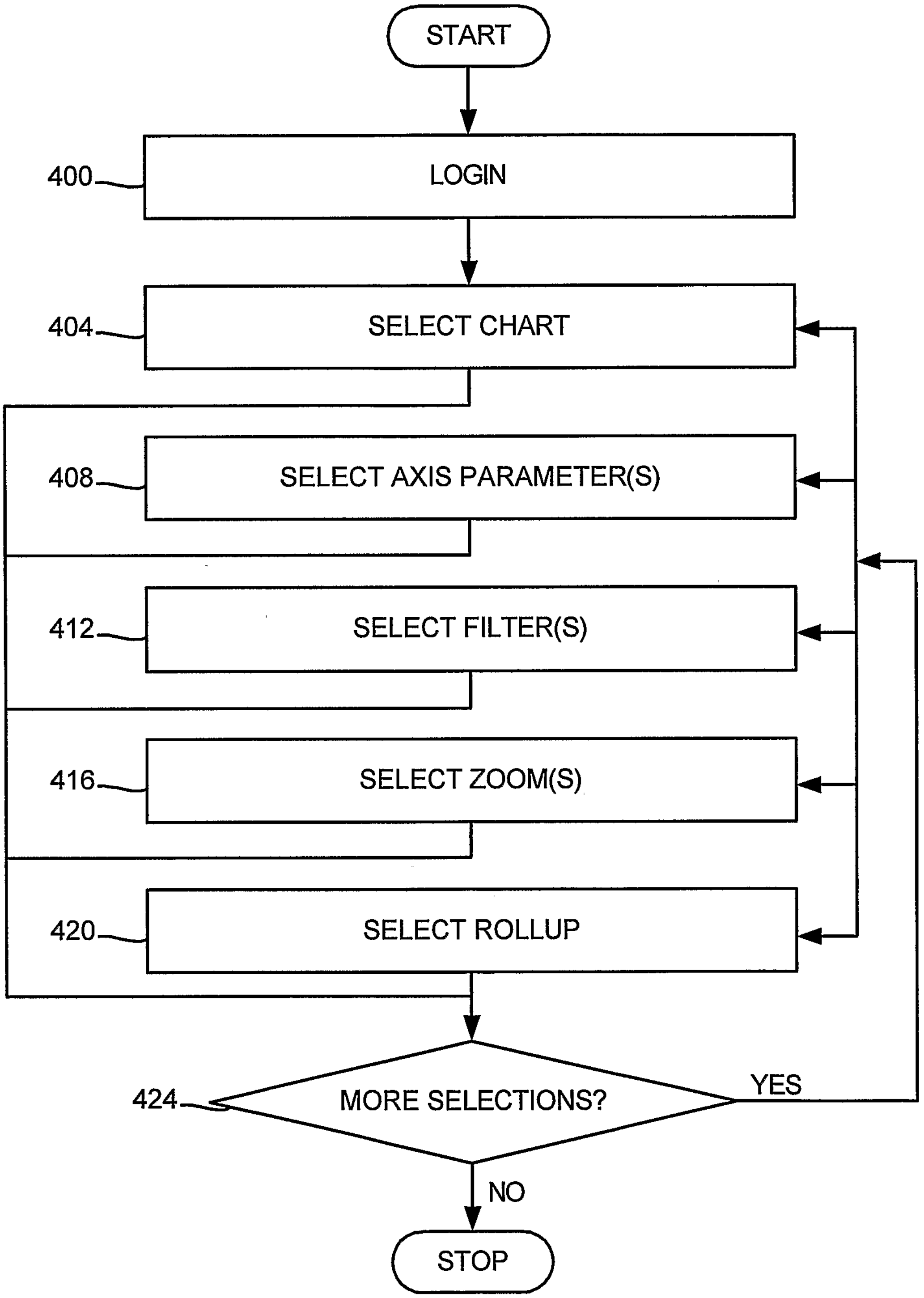

START
400
LOGIN
404
SELECT CHART
408
SELECT AXIS PARAMETER(S)
412
SELECT FILTER(S)
416
SELECT ZOOM(S)
420
SELECT ROLLUP
424
MORE SELECTIONS?
YES
NO
STOP

FIG. 4

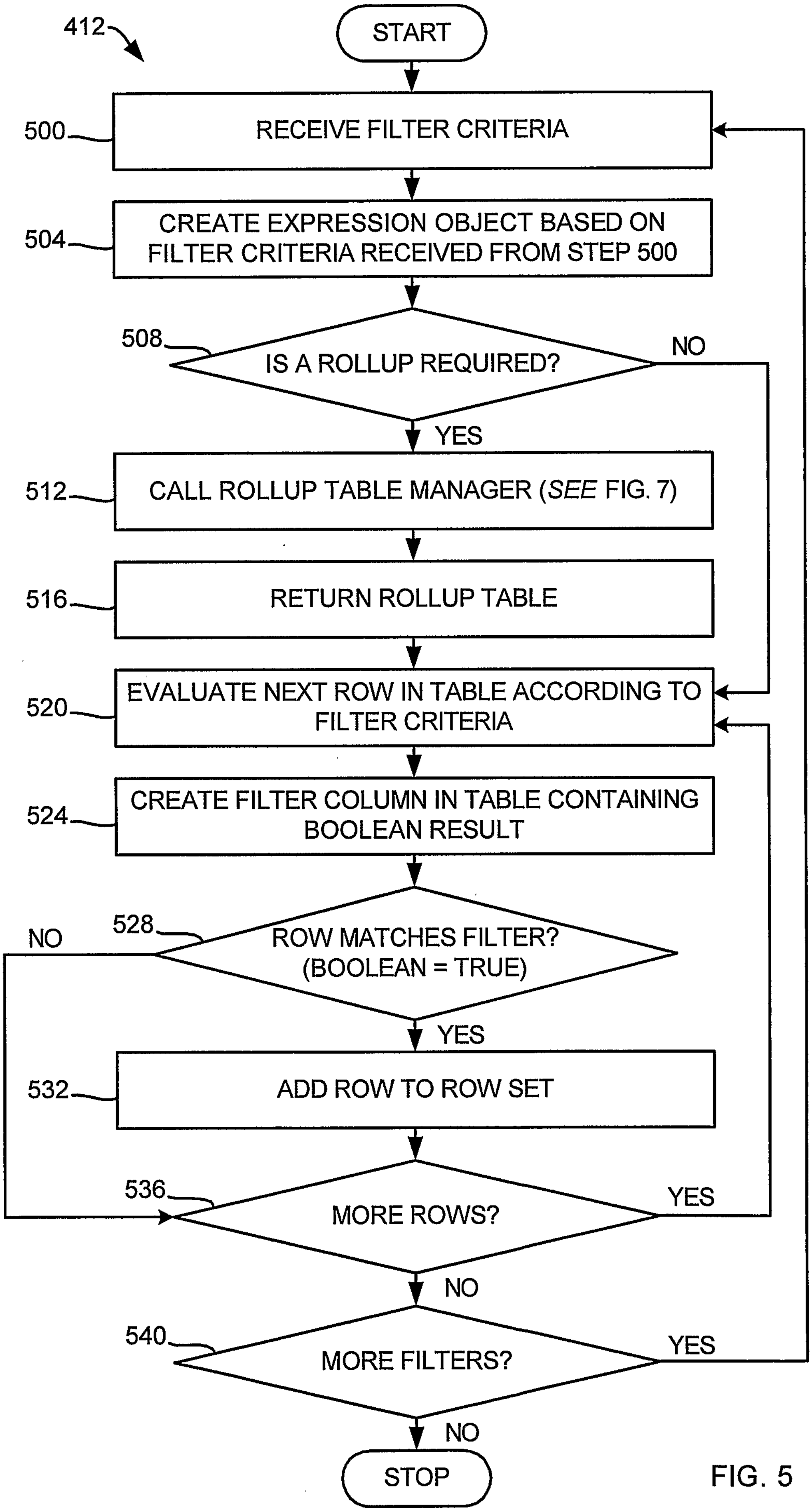

412
START
500
RECEIVE FILTER CRITERIA
504
CREATE EXPRESSION OBJECT BASED ON FILTER CRITERIA RECEIVED FROM STEP 500
508
IS A ROLLUP REQUIRED?
NO
YES
512
CALL ROLLUP TABLE MANAGER (*SEE* FIG. 7)
516
RETURN ROLLUP TABLE
520
EVALUATE NEXT ROW IN TABLE ACCORDING TO FILTER CRITERIA
524
CREATE FILTER COLUMN IN TABLE CONTAINING BOOLEAN RESULT
528
ROW MATCHES FILTER? (BOOLEAN = TRUE)
NO
YES
532
ADD ROW TO ROW SET
536
MORE ROWS?
YES
NO
540
MORE FILTERS?
YES
NO
STOP

FIG. 5

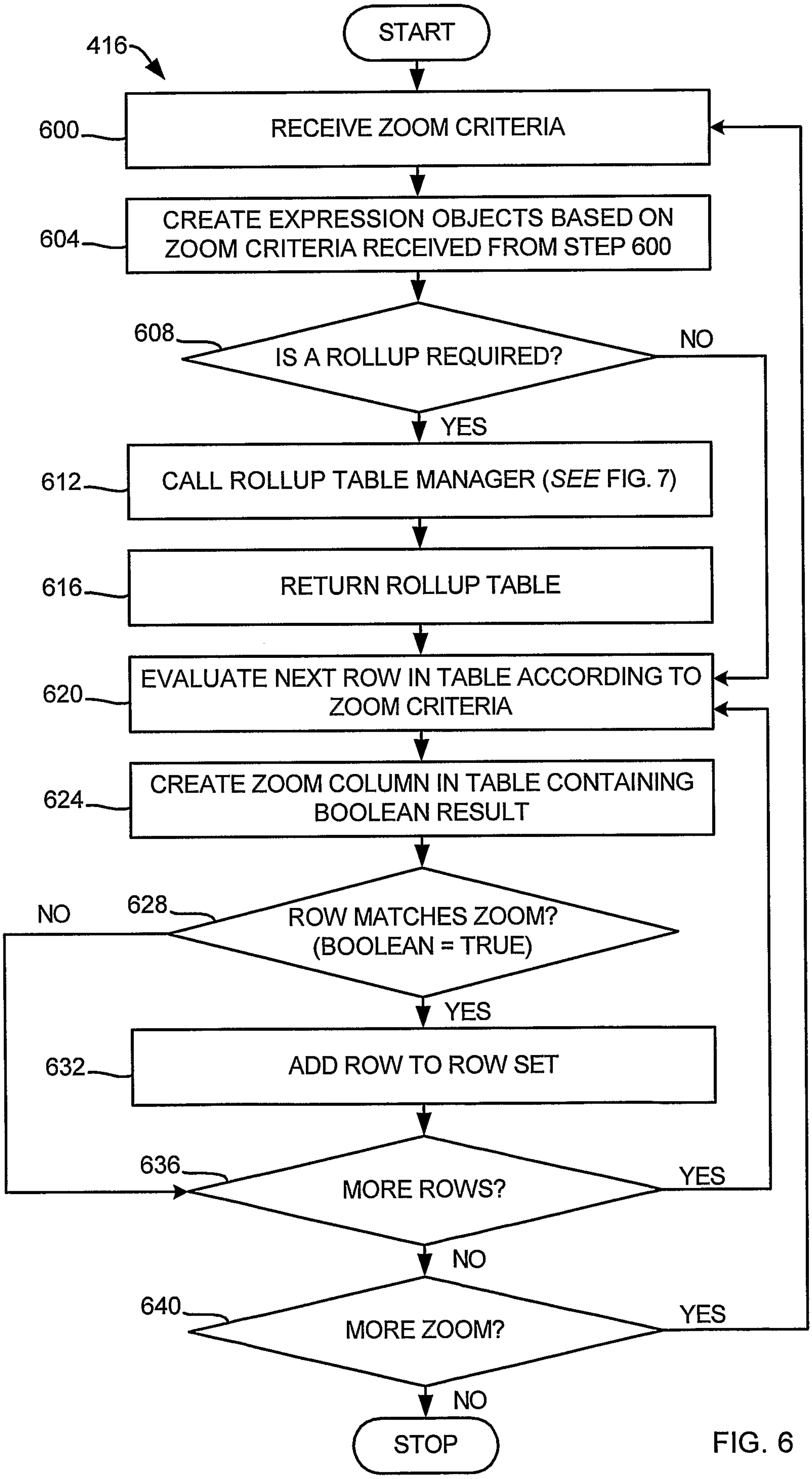

START
416
600
RECEIVE ZOOM CRITERIA
604
CREATE EXPRESSION OBJECTS BASED ON ZOOM CRITERIA RECEIVED FROM STEP 600
608
IS A ROLLUP REQUIRED?
NO
YES
612
CALL ROLLUP TABLE MANAGER (SEE FIG. 7)
616
RETURN ROLLUP TABLE
620
EVALUATE NEXT ROW IN TABLE ACCORDING TO ZOOM CRITERIA
624
CREATE ZOOM COLUMN IN TABLE CONTAINING BOOLEAN RESULT
NO
628
ROW MATCHES ZOOM? (BOOLEAN = TRUE)
YES
632
ADD ROW TO ROW SET
636
MORE ROWS?
YES
NO
640
MORE ZOOM?
YES
NO
STOP

FIG. 6

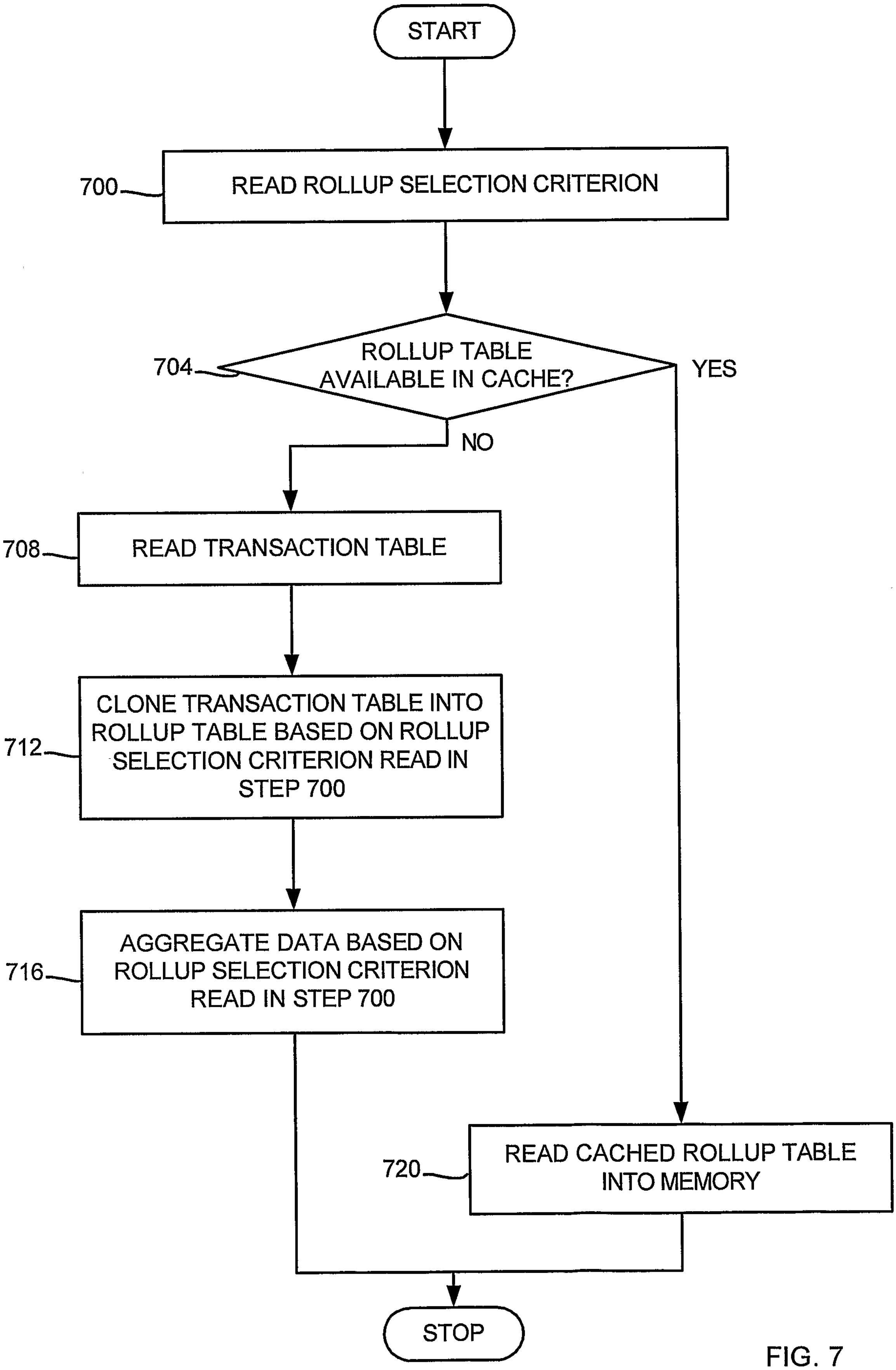

START
700
READ ROLLUP SELECTION CRITERION
704
ROLLUP TABLE AVAILABLE IN CACHE?
YES
NO
708
READ TRANSACTION TABLE
712
CLONE TRANSACTION TABLE INTO ROLLUP TABLE BASED ON ROLLUP SELECTION CRITERION READ IN STEP 700
716
AGGREGATE DATA BASED ON ROLLUP SELECTION CRITERION READ IN STEP 700
720
READ CACHED ROLLUP TABLE INTO MEMORY
STOP

FIG. 7