



US 20170249066A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2017/0249066 A1**

**Zeldis et al.** (43) **Pub. Date: Aug. 31, 2017**

(54) **GRAPHICAL USER INTERFACE DISPLAY EFFECTS FOR A COMPUTER DISPLAY SCREEN**

**Related U.S. Application Data**

(63) Continuation of application No. 15/155,007, filed on May 14, 2016, now Pat. No. 9,613,018.

(71) Applicant: **Walleye Software, LLC**, Plymouth, MN (US)

(60) Provisional application No. 62/161,813, filed on May 14, 2015.

(72) Inventors: **Mark Zeldis**, Randolph, NJ (US); **Ryan Caudy**, New York, NY (US); **David R. Kent, IV**, Colorado Springs, CO (US); **Charles Wright**, Cortlandt Manor, NY (US); **Radu Teodorescu**, New York, NY (US); **Nathaniel J. Dorfman**, Sandy, UT (US)

**Publication Classification**

(51) **Int. Cl.**  
*G06F 3/0483* (2006.01)  
*G06F 3/0485* (2006.01)  
*G06F 17/30* (2006.01)  
(52) **U.S. Cl.**  
CPC ..... *G06F 3/0483* (2013.01); *G06F 17/30864* (2013.01); *G06F 3/0485* (2013.01)

(21) Appl. No.: **15/428,145**

(57) **ABSTRACT**

(22) Filed: **Feb. 8, 2017**

Described are methods, systems and computer readable media for GUI display effect formatting.

500 GUI display with hidden color GUI display effect column

502 Content with color formula for AvgPrice < \$90 (RED) & > \$100 (GREEN)			504 Hidden Column
SYMBOL	MONTH	AVGPRICE	AVGPRICE ROW COLOR
AAPL	JAN	\$99.83	NULL
AAPL	FEB	\$101.96	GREEN
AAPL	MAR	\$103.54	GREEN
AAPL	APR	\$98.23	NULL
AAPL	MAY	\$92.87	NULL
AAPL	JUN	\$89.13	RED
AAPL	JUL	\$90.01	NULL
AAPL	AUG	\$88.56	RED
AAPL	SEP	\$96.72	NULL
AAPL	OCT	\$103.79	GREEN
AAPL	NOV	\$109.33	GREEN
AAPL	DEC	\$115.92	GREEN

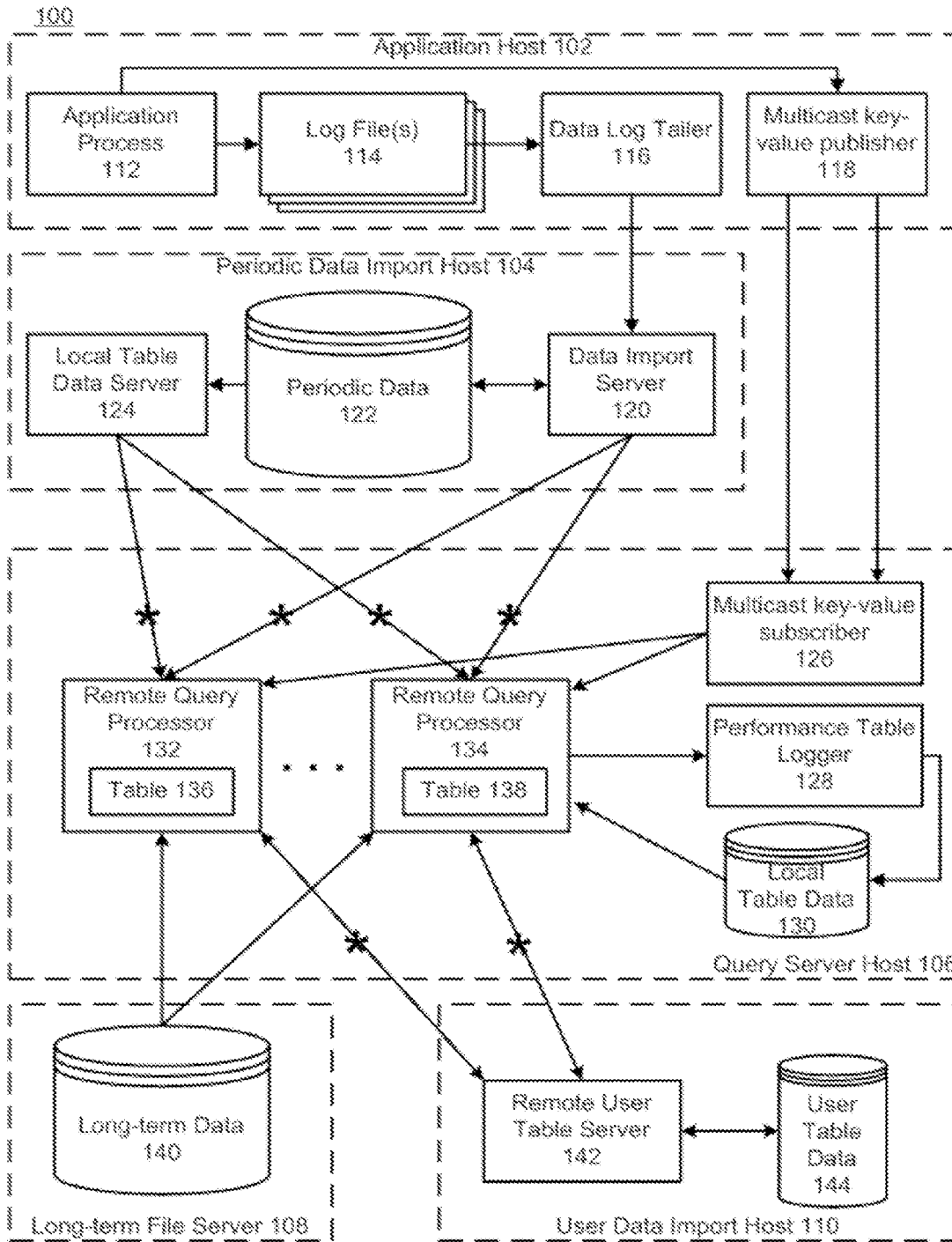


FIG. 1

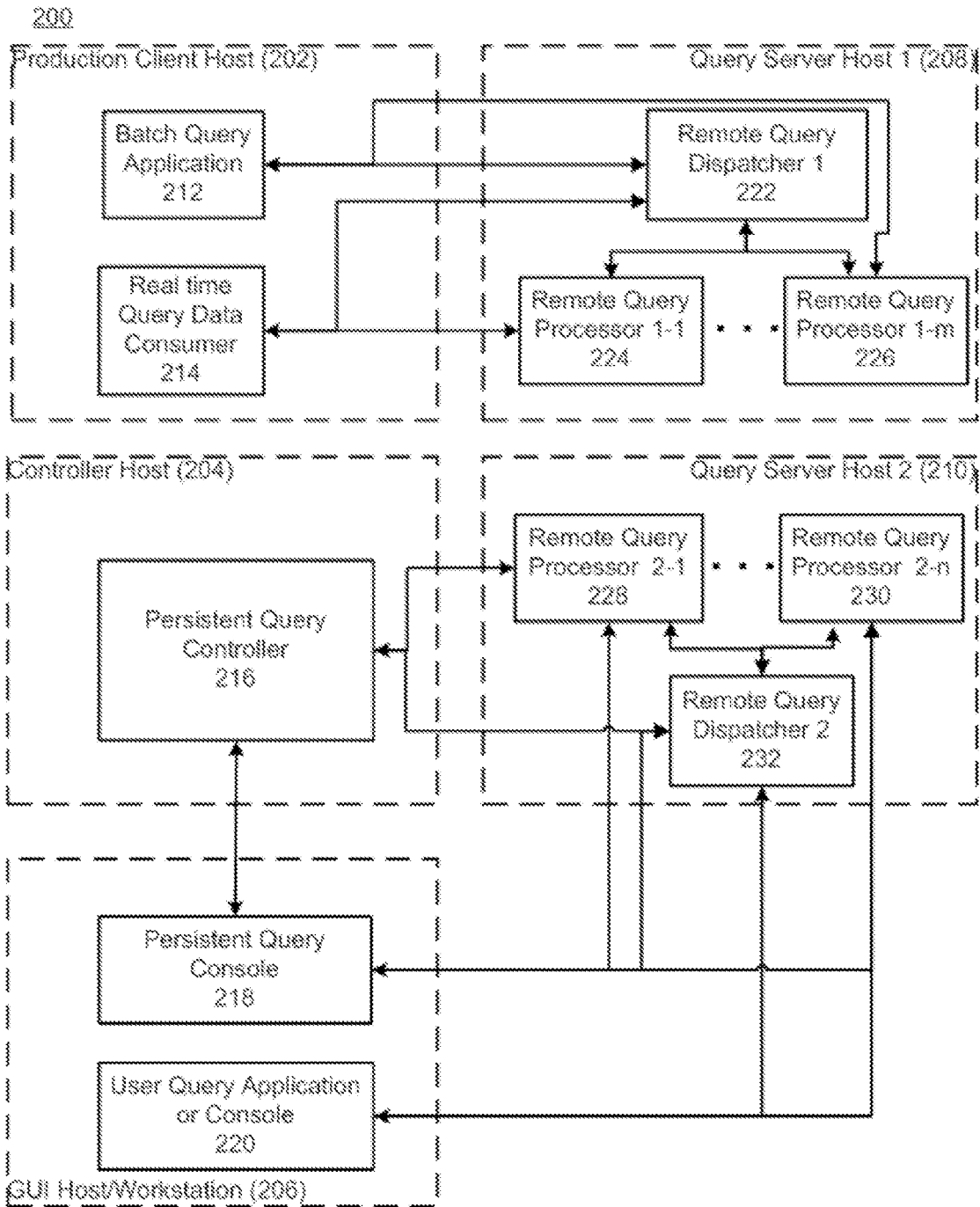


FIG. 2

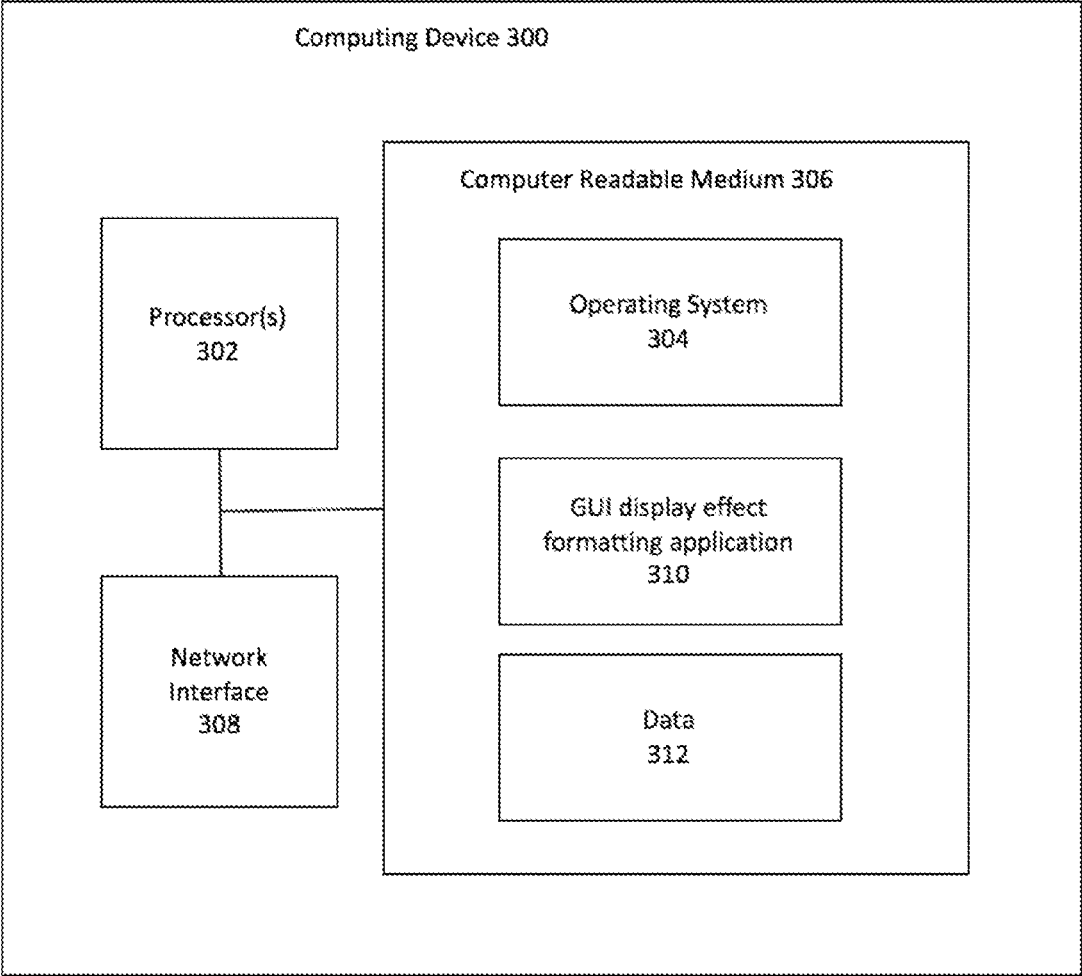


FIG. 3

400 GUI display with hidden color GUI display effect column

402 Content with color formula for AvgPrice < \$90 (RED) & > \$100 (GREEN)			404 Hidden Column
SYMBOL	MONTH	AVGPRICE	AVGPRICE CELL COLOR
AAPL	JAN	\$99.83	NULL
AAPL	FEB	\$101.96	GREEN
AAPL	MAR	\$103.54	GREEN
AAPL	APR	\$98.23	NULL
AAPL	MAY	\$92.87	NULL
AAPL	JUN	\$89.13	RED
AAPL	JUL	\$90.01	NULL
AAPL	AUG	\$88.56	RED
AAPL	SEP	\$96.72	NULL
AAPL	OCT	\$103.79	GREEN
AAPL	NOV	\$109.33	GREEN
AAPL	DEC	\$115.92	GREEN

FIG. 4

450 GUI visible content with a cell color GUI display effect applied

SYMBOL	MONTH	AVGPRICE
AAPL	JAN	\$99.83
AAPL	FEB	\$101.96
AAPL	MAR	\$103.54
AAPL	APR	\$98.23
AAPL	MAY	\$92.87
AAPL	JUN	\$89.13
AAPL	JUL	\$90.01
AAPL	AUG	\$88.56
AAPL	SEP	\$96.72
AAPL	OCT	\$103.79
AAPL	NOV	\$109.33
AAPL	DEC	\$115.92

FIG. 4A

500 GUI display with hidden color GUI display effect column

502 Content with color formula for AvgPrice < \$90 (RED) & > \$100 (GREEN)			504 Hidden Column
SYMBOL	MONTH	AVGPRICE	AVGPRICE ROW COLOR
AAPL	JAN	\$99.83	NULL
AAPL	FEB	\$101.96	GREEN
AAPL	MAR	\$103.54	GREEN
AAPL	APR	\$98.23	NULL
AAPL	MAY	\$92.87	NULL
AAPL	JUN	\$89.13	RED
AAPL	JUL	\$90.01	NULL
AAPL	AUG	\$88.56	RED
AAPL	SEP	\$96.72	NULL
AAPL	OCT	\$103.79	GREEN
AAPL	NOV	\$109.33	GREEN
AAPL	DEC	\$115.92	GREEN

FIG. 5

550 GUI visible content with a row color GUI display effect applied

SYMBOL	MONTH	AVGPRICE	
AAPL	JAN	\$99.83	→ GREEN 552
AAPL	FEB	\$101.96	→ GREEN 554
AAPL	MAR	\$103.54	→ GREEN 554
AAPL	APR	\$98.23	
AAPL	MAY	\$92.87	
AAPL	JUN	\$89.13	→ RED 556
AAPL	JUL	\$90.01	
AAPL	AUG	\$88.56	→ RED 558
AAPL	SEP	\$96.72	
AAPL	OCT	\$103.79	→ GREEN 560
AAPL	NOV	\$109.33	→ GREEN 562
AAPL	DEC	\$115.92	→ GREEN 564

FIG. 5A



600

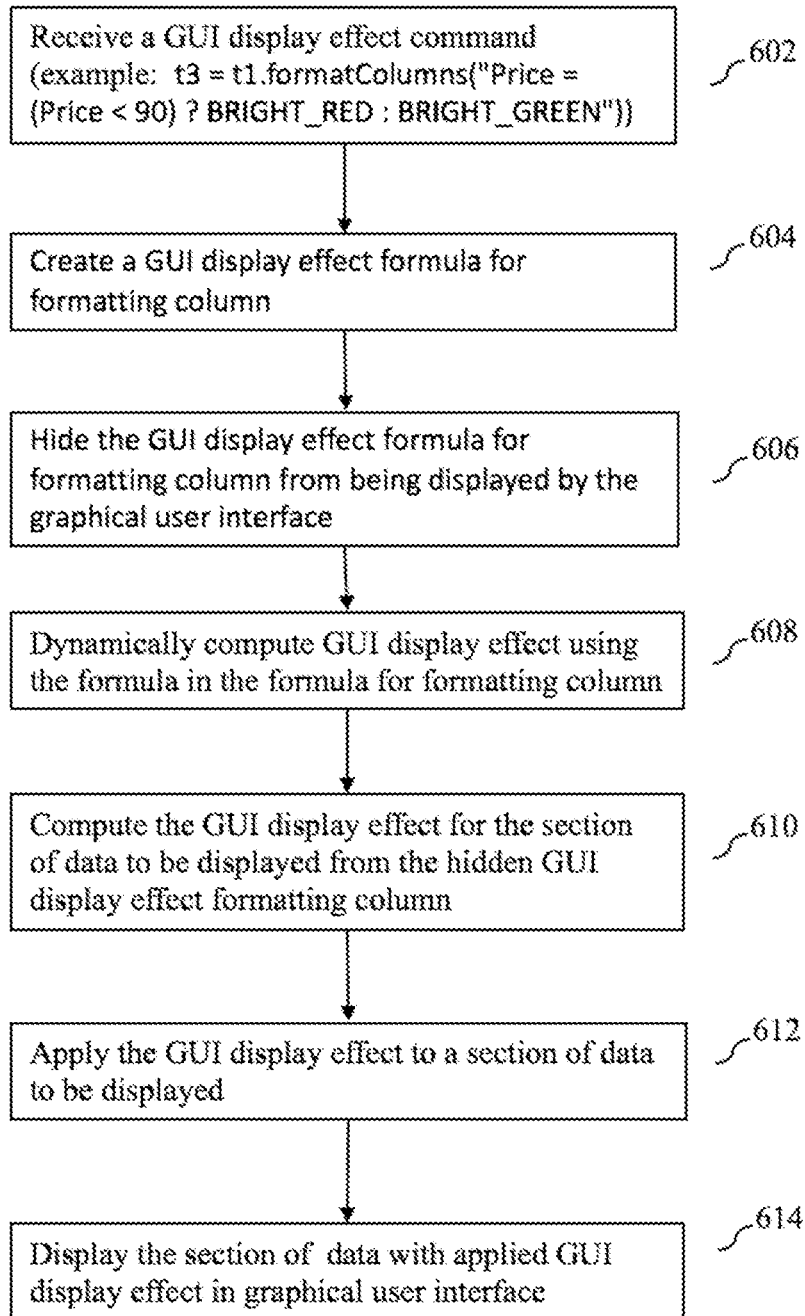


FIG. 6

700 GUI display with hidden font GUI display effect column

702 Content with font formula for AvgPrice > \$100 (ITALICS)			704 Hidden Column
SYMBOL	MONTH	AVGPRICE	AVGPRICE CELL FONT
AAPL	JAN	\$99.83	NULL
AAPL	FEB	\$101.96	ITALICS
AAPL	MAR	\$103.54	ITALICS
AAPL	APR	\$98.23	NULL
AAPL	MAY	\$92.87	NULL
AAPL	JUN	\$89.13	NULL
AAPL	JUL	\$90.01	NULL
AAPL	AUG	\$88.56	NULL
AAPL	SEP	\$96.72	NULL
AAPL	OCT	\$103.79	ITALICS
AAPL	NOV	\$109.33	ITALICS
AAPL	DEC	\$115.92	ITALICS

FIG. 7

750 GUI visible content with a cell font GUI display effect applied

SYMBOL	MONTH	AVGPRICE
AAPL	JAN	\$99.83
AAPL	FEB	<i>\$101.96</i>
AAPL	MAR	<i>\$103.54</i>
AAPL	APR	\$98.23
AAPL	MAY	\$92.87
AAPL	JUN	\$89.13
AAPL	JUL	\$90.01
AAPL	AUG	\$88.56
AAPL	SEP	\$96.72
AAPL	OCT	<i>\$103.79</i>
AAPL	NOV	<i>\$109.33</i>
AAPL	DEC	<i>\$115.92</i>

FIG. 7A

800 GUI display with hidden font GUI display effect column

802 Content with font formula for AvgPrice > \$100 (ITALICS)			804 Hidden Column
SYMBOL	MONTH	AVGPRICE	AVGPRICE ROW FONT
AAPL	JAN	\$99.83	NULL
AAPL	FEB	\$101.96	ITALICS
AAPL	MAR	\$103.54	ITALICS
AAPL	APR	\$98.23	NULL
AAPL	MAY	\$92.87	NULL
AAPL	JUN	\$89.13	NULL
AAPL	JUL	\$90.01	NULL
AAPL	AUG	\$88.56	NULL
AAPL	SEP	\$96.72	NULL
AAPL	OCT	\$103.79	ITALICS
AAPL	NOV	\$109.33	ITALICS
AAPL	DEC	\$115.92	ITALICS

FIG. 8

850 GUI visible content with a row color GUI display effect applied

SYMBOL	MONTH	AVGPRICE
AAPL	JAN	\$99.83
<i>AAPL</i>	<i>FEB</i>	<i>\$101.96</i>
<i>AAPL</i>	<i>MAR</i>	<i>\$103.54</i>
AAPL	APR	\$98.23
AAPL	MAY	\$92.87
AAPL	JUN	\$89.13
AAPL	JUL	\$90.01
AAPL	AUG	\$88.56
AAPL	SEP	\$96.72
<i>AAPL</i>	<i>OCT</i>	<i>\$103.79</i>
<i>AAPL</i>	<i>NOV</i>	<i>\$109.33</i>
<i>AAPL</i>	<i>DEC</i>	<i>\$115.92</i>

FIG. 8A

900

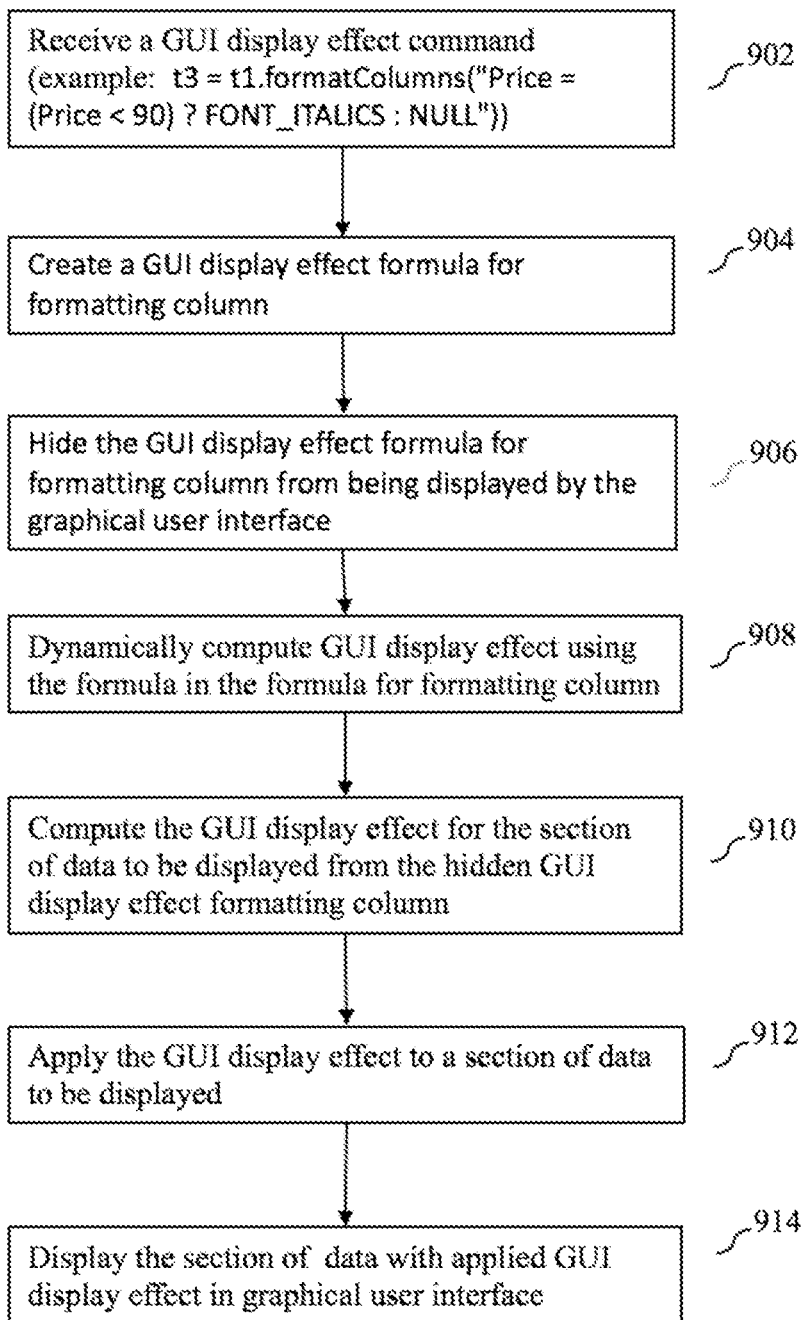


FIG. 9

**GRAPHICAL USER INTERFACE DISPLAY  
EFFECTS FOR A COMPUTER DISPLAY  
SCREEN**

**[0001]** This application claims the benefit of U.S. Provisional Application No. 62/161,813, entitled "Computer Data System" and filed on May 14, 2015, which is incorporated herein by reference in its entirety.

**[0002]** Embodiments relate generally to computer data systems, and more particularly, to methods, systems and computer readable media for providing GUI display effect formatting for the display of data based on the content of the data.

**[0003]** Graphical user interfaces (GUI) can display a multitude of data in many different formats including a table row and column format or a spreadsheet cell format. The larger the screen and the smaller the font size, the more rows and cells that can be simultaneously displayed on the GUI. Rows and cells can also be presented in a rolling screen format with the addition of fresh row or cell data causing older row or cell data to roll off the screen and out of sight. Under such conditions, important or crucial data can be missed or overlooked by a user. In addition to rolling screen data, data sources can also be dynamic. Dynamic data sources can have data added or modified frequently. A change to a dynamic data source that is being displayed and updated in real-time intervals on the screen can cause a change to an important or critical value used in decision making that might be overlooked by a user.

**[0004]** Embodiments were conceived in light of the above mentioned needs, problems and/or limitations, among other things.

**[0005]** Some implementations can include a system for formatting data for display on a computer display screen, the system comprising one or more hardware processors. The system can also include a computer display screen configured to display data via a graphical user interface (GUI). The system can further include a computer readable data storage device coupled to the one or more hardware processors, the computer readable data storage device having stored thereon software instructions that, when executed by the one or more hardware processors, cause the one or more hardware processors to perform operations. The operations can include receiving a digital signal containing a GUI display effect formatting command for data stored in a computer medium. The operations can also include creating a GUI display effect formula for a formatting column. The operations can include hiding the formatting column from being displayed by the graphical user interface on the computer display screen. The operations can also include dynamically computing a GUI display effect using the GUI display effect formula for a formatting column. The operations can further include choosing a section of data to be displayed on the computer display screen via the graphical user interface. The operations can include computing the GUI display effect for the section of data to be displayed on the computer display screen via the graphical interface using the hidden formatting column. The operations can also include applying the computed GUI display effect to the section of data to be displayed. The operations can include displaying on the computer display screen the section of data with the applied computed GUI display effect.

**[0006]** The operations can further include applying the GUI display effect formatting by applying a formula to associated data in a same row of data.

**[0007]** The operations can include wherein applying the GUI display effect formatting includes applying a GUI display effect to at least one of an entire row of data, an entire column of data, an entire table, a tab, a browser, and a graphical user interface.

**[0008]** The operations can also include wherein the section of data to be displayed on the graphical interface using the hidden formatting column is determined by scrolling.

**[0009]** The operations can also include receiving updates to the section of data to be displayed on the graphical user interface. The operations can include using a processor, dynamically computing a GUI display effect for the updates to the section of data using the GUI display effect formula for a formatting column. The operations can further include choosing a section of data to be displayed on the graphical user interface. The operations can include computing the GUI display effect for the section of data to be displayed on the graphical interface using the hidden formatting column. The operations can include applying the computed GUI display effect to the section of data to be displayed. The operations can also include displaying the section of data with the applied computed GUI display effect.

**[0010]** The operations can include wherein computing the GUI display effect occurs prior to displaying the section of data.

**[0011]** The operations can include wherein computing the GUI display effect occurs at the time of displaying the section of data.

**[0012]** Some implementations can include a method for formatting data displayed in a graphical user interface (GUI) comprising receiving a GUI display effect formatting command for data stored in a computer medium. The method can also include creating a GUI display effect formula for a formatting column. The method can further include hiding the formatting column from being displayed by the graphical user interface. The method can include using a processor, dynamically computing a GUI display effect using the GUI display effect formula for a formatting column. The method can also include choosing a section of data to be displayed on the graphical user interface. The method can include computing the GUI display effect for the section of data to be displayed on the graphical interface using the hidden formatting column. The method can also include applying the computed GUI display effect to the section of data to be displayed. The method can further include displaying the section of data with the applied computed GUI display effect.

**[0013]** The method can further include applying the GUI display effect formatting by applying a formula to associated data in a same row of data.

**[0014]** The method can include wherein applying the GUI display effect formatting includes applying a GUI display effect to at least one of an entire row of data, an entire column of data, an entire table, a tab, a browser, and a graphical user interface.

**[0015]** The method can also include wherein the section of data to be displayed on the graphical interface using the hidden formatting column is determined by scrolling.

**[0016]** The method can include receiving updates to the section of data to be displayed on the graphical user interface. The method can also include using a processor, dynamically computing a GUI display effect for the updates

to the section of data using the GUI display effect formula for a formatting column. The method can further include choosing a section of data to be displayed on the graphical user interface. The method can include computing the GUI display effect for the section of data to be displayed on the graphical interface using the hidden formatting column. The method can include applying the computed GUI display effect to the section of data to be displayed. The method can also include displaying the section of data with the applied computed GUI display effect.

**[0017]** The method can include wherein computing the GUI display effect occurs prior to displaying the section of data.

**[0018]** Some implementations can include a nontransitory computer readable medium having stored thereon software instructions that, when executed by one or more processors, cause the one or more processors to perform operations. The operations can include receiving a digital signal containing a GUI display effect formatting command for data stored in a computer medium. The operations can also include creating a GUI display effect formula for a formatting column. The operations can further include hiding the formatting column from being displayed by the graphical user interface on the computer display screen. The operations can include dynamically computing a GUI display effect using the GUI display effect formula for a formatting column. The operations can also include choosing a section of data to be displayed on the computer display screen via the graphical user interface. The operations can include computing the GUI display effect for the section of data to be displayed on the the computer display screen via the graphical interface using the hidden formatting column. The operations can also include applying the computed GUI display effect to the section of data to be displayed. The operations can include displaying on the computer display screen the section of data with the applied computed GUI display effect.

**[0019]** The operations can include applying the GUI display effect formatting by applying a formula to associated data in a same row of data.

**[0020]** The operations can include wherein applying the GUI display effect formatting includes applying a GUI display effect to at least one of an entire row of data, an entire column of data, an entire table, a tab, a browser, and a graphical user interface.

**[0021]** The operations can include wherein the the section of data to be displayed on the graphical interface using the hidden formatting column is determined by scrolling.

**[0022]** The operations can include receiving updates to the section of data to be displayed on the graphical user interface. The operations can also include using a processor, dynamically computing a GUI display effect for the updates to the section of data using the GUI display effect formula for a formatting column. The operations can further include choosing a section of data to be displayed on the graphical user interface. The operations can include computing the GUI display effect for the section of data to be displayed on the graphical interface using the hidden formatting column. The operations can also include applying the computed GUI display effect to the section of data to be displayed. The operations can include displaying the section of data with the applied computed GUI display effect.

**[0023]** The operations can include wherein computing the GUI display effect occurs prior to displaying the section of data.

**[0024]** The operations can also include wherein computing the GUI display effect occurs at the time of displaying the section of data.

#### BRIEF DESCRIPTION OF THE DRAWINGS

**[0025]** FIG. 1 is a diagram of an example computer data system showing an example data distribution configuration in accordance with some implementations.

**[0026]** FIG. 2 is a diagram of an example computer data system showing an example administration/process control arrangement in accordance with some implementations.

**[0027]** FIG. 3 is a diagram of an example computing device configured for GUI display effect format processing in accordance with some implementations.

**[0028]** FIG. 4 is a diagram of an example of a GUI display with hidden GUI display effect column in accordance with some implementations.

**[0029]** FIG. 4A is a diagram of an example GUI visible content with a cell GUI display effect applied in accordance with some implementations.

**[0030]** FIG. 5 is a diagram of an example of a GUI display with hidden GUI display effect column in accordance with some implementations.

**[0031]** FIG. 5A is a diagram of an example GUI visible content with a row GUI display effect applied in accordance with some implementations.

**[0032]** FIG. 6 is a diagram of an example flow chart for determining and displaying a GUI display effect in accordance with some implementations.

**[0033]** FIG. 7 is a diagram of an example of a GUI display with hidden GUI display effect column in accordance with some implementations.

**[0034]** FIG. 7A is a diagram of an example GUI visible content with a cell GUI display effect applied in accordance with some implementations.

**[0035]** FIG. 8 is a diagram of an example of a GUI display with hidden GUI display effect column in accordance with some implementations.

**[0036]** FIG. 8A is a diagram of an example GUI visible content with a row GUI display effect applied in accordance with some implementations.

**[0037]** FIG. 9 is a diagram of an example flow chart for determining and displaying a GUI display effect in accordance with some implementations.

#### DETAILED DESCRIPTION

**[0038]** Reference is made herein to the Java programming language, Java classes, Java bytecode and the Java Virtual Machine (JVM) for purposes of illustrating example implementations. It will be appreciated that implementations can include other programming languages (e.g., groovy, Scala, R, Go, etc.), other programming language structures as an alternative to or in addition to Java classes (e.g., other language classes, objects, data structures, program units, code portions, script portions, etc.), other types of bytecode, object code and/or executable code, and/or other virtual machines or hardware implemented machines configured to execute a data system query.

**[0039]** FIG. 1 is a diagram of an example computer data system and network 100 showing an example data distribution configuration in accordance with some implementations. In particular, the system 100 includes an application host 102, a periodic data import host 104, a query server host



**106**, a long-term file server **108**, and a user data import host **110**. While tables are used as an example data object in the description below, it will be appreciated that the data system described herein can also process other data objects such as mathematical objects (e.g., a singular value decomposition of values in a given range of one or more rows and columns of a table), TableMap objects, etc. A TableMap object provides the ability to lookup a Table by some key. This key represents a unique value (or unique tuple of values) from the columns aggregated on in a `byExternal()` statement execution, for example. A TableMap object is can be the result of a `byExternal()` statement executed as part of a query. It will also be appreciated that the configurations shown in FIGS. 1 and 2 are for illustration purposes and in a given implementation each data pool (or data store) may be directly attached or may be managed by a file server.

[0040] The application host **102** can include one or more application processes **112**, one or more log files **114** (e.g., sequential, row-oriented log files), one or more data log tailers **116** and a multicast key-value publisher **118**. The periodic data import host **104** can include a local table data server, direct or remote connection to a periodic table data store **122** (e.g., a column-oriented table data store) and a data import server **120**. The query server host **106** can include a multicast key-value subscriber **126**, a performance table logger **128**, local table data store **130** and one or more remote query processors (**132**, **134**) each accessing one or more respective tables (**136**, **138**). The long-term file server **108** can include a long-term data store **140**. The user data import host **110** can include a remote user table server **142** and a user table data store **144**. Row-oriented log files and column-oriented table data stores are discussed herein for illustration purposes and are not intended to be limiting. It will be appreciated that log files and/or data stores may be configured in other ways. In general, any data stores discussed herein could be configured in a manner suitable for a contemplated implementation.

[0041] In operation, the input data application process **112** can be configured to receive input data from a source (e.g., a securities trading data source), apply schema-specified, generated code to format the logged data as it's being prepared for output to the log file **114** and store the received data in the sequential, row-oriented log file **114** via an optional data logging process. In some implementations, the data logging process can include a daemon, or background process task, that is configured to log raw input data received from the application process **112** to the sequential, row-oriented log files on disk and/or a shared memory queue (e.g., for sending data to the multicast publisher **118**). Logging raw input data to log files can additionally serve to provide a backup copy of data that can be used in the event that downstream processing of the input data is halted or interrupted or otherwise becomes unreliable.

[0042] A data log tailer **116** can be configured to access the sequential, row-oriented log file(s) **114** to retrieve input data logged by the data logging process. In some implementations, the data log tailer **116** can be configured to perform strict byte reading and transmission (e.g., to the data import server **120**). The data import server **120** can be configured to store the input data into one or more corresponding data stores such as the periodic table data store **122** in a column-oriented configuration. The periodic table data store **122** can be used to store data that is being received within a time period (e.g., a minute, an hour, a day, etc.) and which may

be later processed and stored in a data store of the long-term file server **108**. For example, the periodic table data store **122** can include a plurality of data servers configured to store periodic securities trading data according to one or more characteristics of the data (e.g., a data value such as security symbol, the data source such as a given trading exchange, etc.).

[0043] The data import server **120** can be configured to receive and store data into the periodic table data store **122** in such a way as to provide a consistent data presentation to other parts of the system. Providing/ensuring consistent data in this context can include, for example, recording logged data to a disk or memory, ensuring rows presented externally are available for consistent reading (e.g., to help ensure that if the system has part of a record, the system has all of the record without any errors), and preserving the order of records from a given data source. If data is presented to clients, such as a remote query processor (**132**, **134**), then the data may be persisted in some fashion (e.g., written to disk).

[0044] The local table data server **124** can be configured to retrieve data stored in the periodic table data store **122** and provide the retrieved data to one or more remote query processors (**132**, **134**) via an optional proxy.

[0045] The remote user table server (RUTS) **142** can include a centralized consistent data writer, as well as a data server that provides processors with consistent access to the data that it is responsible for managing. For example, users can provide input to the system by writing table data that is then consumed by query processors.

[0046] The remote query processors (**132**, **134**) can use data from the data import server **120**, local table data server **124** and/or from the long-term file server **108** to perform queries. The remote query processors (**132**, **134**) can also receive data from the multicast key-value subscriber **126**, which receives data from the multicast key-value publisher **118** in the application host **102**. The performance table logger **128** can log performance information about each remote query processor and its respective queries into a local table data store **130**. Further, the remote query processors can also read data from the RUTS, from local table data written by the performance logger, or from user table data read over NFS.

[0047] It will be appreciated that the configuration shown in FIG. 1 is a typical example configuration that may be somewhat idealized for illustration purposes. An actual configuration may include one or more of each server and/or host type. The hosts/servers shown in FIG. 1 (e.g., **102-110**, **120**, **124** and **142**) may each be separate or two or more servers may be combined into one or more combined server systems. Data stores can include local/remote, shared/isolated and/or redundant. Any table data may flow through optional proxies indicated by an asterisk on certain connections to the remote query processors. Also, it will be appreciated that the term "periodic" is being used for illustration purposes and can include, but is not limited to, data that has been received within a given time period (e.g., millisecond, second, minute, hour, day, week, month, year, etc.) and which has not yet been stored to a long-term data store (e.g., **140**).

[0048] FIG. 2 is a diagram of an example computer data system **200** showing an example administration/process control arrangement in accordance with some implementations. The system **200** includes a production client host **202**, a controller host **204**, a GUI host or workstation **206**, and

query server hosts **208** and **210**. It will be appreciated that there may be one or more of each of **202-210** in a given implementation.

**[0049]** The production client host **202** can include a batch query application **212** (e.g., a query that is executed from a command line interface or the like) and a real time query data consumer process **214** (e.g., an application that connects to and listens to tables created from the execution of a separate query). The batch query application **212** and the real time query data consumer **214** can connect to a remote query dispatcher **222** and one or more remote query processors (**224**, **226**) within the query server host **1 208**.

**[0050]** The controller host **204** can include a persistent query controller **216** configured to connect to a remote query dispatcher **232** and one or more remote query processors **228-230**. In some implementations, the persistent query controller **216** can serve as the “primary client” for persistent queries and can request remote query processors from dispatchers, and send instructions to start persistent queries. For example, a user can submit a query to **216**, and **216** starts and runs the query every day. In another example, a securities trading strategy could be a persistent query. The persistent query controller can start the trading strategy query every morning before the market opened, for instance. It will be appreciated that **216** can work on times other than days. In some implementations, the controller may require its own clients to request that queries be started, stopped, etc. This can be done manually, or by scheduled (e.g., cron) jobs. Some implementations can include “advanced scheduling” (e.g., auto-start/stop/restart, time-based repeat, etc.) within the controller.

**[0051]** The GUI/host workstation can include a user console **218** and a user query application **220**. The user console **218** can be configured to connect to the persistent query controller **216**. The user query application **220** can be configured to connect to one or more remote query dispatchers (e.g., **232**) and one or more remote query processors (**228**, **230**).

**[0052]** FIG. **3** is a diagram of an example computing device **300** in accordance with at least one implementation. The computing device **300** includes one or more processors **302**, operating system **304**, computer readable medium **306** and network interface **308**. The memory **306** can include a GUI display effect formatting application **310** and a data section **312** (e.g., for storing ASTs, precompiled code, etc.).

**[0053]** In operation, the processor **302** may execute the application **310** stored in the memory **306**. The application **310** can include software instructions that, when executed by the processor, cause the processor to perform operations for executing and updating queries in accordance with the present disclosure (e.g., performing one or more of **602-614**, **902-914** described below).

**[0054]** The application program **310** can operate in conjunction with the data section **312** and the operating system **304**.

**[0055]** In general, some implementations can provide a system for determining GUI display effect for displaying data based on the content of the data to be displayed. The system described herein may determine whether a data row or cell should have a particular GUI display effect applied for that data row or cell when displayed if the data content satisfies certain criteria. For example, one criterion could be whether data content includes a value equal to, above, or below a certain threshold value or within or outside of a

range of values. Another criterion could be how recently the data content has changed, how it has changed, or to what extent it has changed. The GUI display effect may be applied to data being displayed or to the background area of a cell, row, column, or area of the GUI containing the data content, for example. In other implementations, the GUI display effect can be applied to table columns or a combination of rows and columns.

**[0056]** It will be appreciated that thresholds and ranges are only exemplary methods for determining display effects. Arbitrary formulas within a data system can be used for computing GUI display effects. Additionally, other columns or rows than the column to be formatted can be used as input to the computation of GUI display effects.

**[0057]** The GUI display effect may use different colors, shading, brightness, flashing, motion, sizing, and/or other visual effect to call attention to or differentiate an area of a display, a browser page or data contained therein. While certain exemplary embodiments are described herein using different colors as a GUI display effect, it is understood that any GUI display effect may supplement or substitute for coloring as a GUI display effect.

**[0058]** The GUI display effect may be applied to a computer screen or monitor such as a CRT, projector, LCD, LED, e-ink display, or OLED screen, for example.

**[0059]** It will be appreciated that the GUI display effect can be applied to any computing device that contains a viewable screen, or projections, such as tablets, smart phones, glasses, heads-up-displays, or the like.

**[0060]** In some implementations, the data source can be a dynamic data source in which the data being displayed on a graphical user interface screen can change as the data source content changes. The system described herein can modify the applied GUI display effect according to dynamic changes to the underlying data used to determine and apply the initial GUI display effect.

**[0061]** FIG. **4** is a diagram of an example Graphical User Interface (GUI) display with an example hidden color GUI display effect column **400** in accordance with some implementations. The display **400** can display content before the application of a color GUI display effect produced by color formula for average prices less than \$90 designated as red and for average prices greater than \$100 designated as green **402**. The content **402** displayed in the GUI display **400** can contain a table with columns for “SYMBOL”, “MONTH”, and “AVGPRICE.” The GUI display can also contain a hidden column **404** that is hidden from being displayed on the GUI display. For example, the hidden column **404** can contain color GUI display effect rows for “AVGPRICE CELL COLOR.” In the example, a processor can receive a format command to apply a red color GUI display effect when AvgPrice is less than \$90 and to apply a green color GUI display effect when the AvgPrice is greater than \$100. The color GUI display effect outcome of the format application can be stored in the hidden column **404**. For example, the hidden column **404** format color GUI display effect for row (AAPL, JAN, \$99.83) is NULL because \$99.83 is greater than \$90 but less than \$100 and no color GUI display effect is provided for this range. In the row (AAPL, FEB, \$101.96), the format color GUI display effect stored in the AVGPRICE CELL COLOR column is GREEN because the AVGPRICE for this row is greater than \$100. In another example row (AAPL, JUN, \$89.13), the format color GUI

display effect stored in the AVGPRICE CELL COLOR column is RED because the AVGPRICE for this row is less than \$90.

**[0062]** FIG. 4A is a diagram of an example GUI visible content with a cell color GUI display effect applied **450** in accordance with some implementations. For example, applying the color GUI display effect contained in the hidden column **404** from the example in FIG. 4, the green (**452, 454, 460, 462, 464**) color GUI display effect from hidden column **404** can be applied to the adjacent cells to the hidden column **404** in the GUI display **450**. Continuing with the example, the red (**456, 458**) color GUI display effect from **404** can be applied to the adjacent cells to the hidden column **404** in the GUI display **450**.

**[0063]** It will be appreciated that the example NULL color GUI display effect values from the hidden column **404** can have no effect on the corresponding cells in **450** and a default format can be rendered.

**[0064]** It will be appreciated that data sources can be static or dynamic. Dynamic data sources can have data added, modified, re-indexed, reordered, or deleted. For example, a dynamic table can have one or more rows added, one or more rows modified, or one or more rows deleted. If a dynamic table has one or more rows added, a GUI display effect will be added for those rows in the hidden column **404**. If a dynamic table has one or more rows modified, the GUI display effect in the hidden column **404** for those rows can change if the data used to determine the GUI display effect is modified enough to move it beyond the existing GUI display effect threshold in the format command. As the dynamic table data is updated on the GUI display screen, the GUI display effect can also be updated on the GUI display screen.

**[0065]** Computer instructions for selecting a possible GUI display effect may be hidden. For example, in a database, the instructions may be in a hidden column or columns. The hidden column or columns may be unavailable to a viewer observing a display showing the GUI display effect on visible columns of the same data source. Alternatively, the hidden column may be available to a user taking a certain action to unhide the column. In that case, the instructions may be configurable by a user of the display. While embodiments are described herein as employing a hidden column, it is understood that multiple hidden columns may be used.

**[0066]** FIG. 5 is a diagram of an example GUI display with a hidden color column **500** in accordance with some implementations. The GUI display **500** can display content before the application of a color GUI display effect produced by color formula for average prices less than \$90 designated as red and for average prices greater than \$100 designated as green **502**. The content **502** displayed in the GUI display **500** can contain a table with columns for "SYMBOL", "MONTH", and "AVGPRICE." The GUI display can also contain one or more hidden columns **504** that is hidden from being displayed on the GUI display. For example, the hidden column **504** can contain color GUI display effect rows for "AVGPRICE ROW COLOR." In the example, a processor can receive a format command to apply a red color GUI display effect when AvgPrice is less than \$90 and to apply a green color GUI display effect when the AvgPrice is greater than \$100. The color GUI display effect outcome of the format application can be stored in the hidden column **504**. For example, the hidden column **504** format color GUI display effect for row (AAPL, JAN, \$99.83) is NULL

because \$99.83 is greater than \$90 but less than \$100 and no color GUI display effect is provided for this range. In the row (AAPL, FEB, \$101.96), the format color GUI display effect stored in the AVGPRICE ROW COLOR column is GREEN because the AVGPRICE for this row is greater than \$100. In another example row (AAPL, JUN, \$89.13), the format color GUI display effect stored in the AVGPRICE ROW COLOR column is RED because the AVGPRICE for this row is less than \$90.

**[0067]** FIG. 5A is a diagram of an example GUI with visible content with a row color GUI display effect applied **550** in accordance with some implementations. For example, applying the color GUI display effect contained in the hidden column **504** from the example in FIG. 5, the green (**552, 554, 560, 562, 564**) color GUI display effect from hidden column **504** can be applied to the corresponding rows in **550**. Continuing with the example, the red (**556, 558**) color GUI display effect from **504** can be applied to the adjacent rows in **550**.

**[0068]** It will be appreciated that the example NULL color GUI display effect values from the hidden column **504** can use the default formatting on the corresponding rows in **550**.

**[0069]** It will be appreciated that data sources can be static or dynamic. Dynamic data sources can have data added, modified, re-indexed, reordered, or deleted. For example, a dynamic table can have one or more rows added, one or more rows modified, or one or more rows deleted. If a dynamic table has one or more rows added, a GUI display effect will be added for those rows in the hidden column **504**. If a dynamic table has one or more rows modified, the color GUI display effect in the hidden column **504** for those rows can change if the data used to determine the color GUI display effect is modified enough to move it beyond the existing color GUI display effect threshold in the format command. As the dynamic table data is updated on the GUI display screen, the GUI display effect can also be updated on the GUI display screen.

**[0070]** FIG. 6 is a diagram of an example flow chart for determining and displaying a GUI display effect in accordance with some implementations. Processing begins at **602**, when the system receives a format command, for example, `t3=t1.formatColumns("Price=(Price<90) ? BRIGHT_RED: BRIGHT_GREEN")` to color format cells in the Price column bright red if Price is less than 90 and bright green for Price column cells greater than or equal to 90.

**[0071]** It will be appreciated the GUI display effect is not limited to coloring. The GUI display effect may use different colors, shading, brightness, flashing, motion, sizing, and/or other visual effect to call attention to or differentiate an area of a display, a browser page or data contained therein. While certain exemplary embodiments are described herein using different colors as a GUI display effect, it is understood that any GUI display effect may supplement or substitute for coloring as a GUI display effect. Processing continues to **604**.

**[0072]** At **604**, a GUI display effect formula for a formatting column is created. Processing continues to **606**.

**[0073]** At **606**, the GUI display effect formula for a formatting column is hidden from being displayed by the graphical user interface. Processing continues to **608**.

**[0074]** At **608**, the GUI display effect is dynamically computed using the formula in the formula for formatting column. Processing continues to **610**.

[0075] At 610, the GUI display effect for the section of data to be displayed is computed from the hidden GUI display effect formatting column. Processing continues to 612.

[0076] At 612, the GUI display effect is applied to a section of data to be displayed. Processing continues to 614.

[0077] At 614, the section of data with the applied GUI display effect is displayed in a graphical user interface.

[0078] FIG. 7 is a diagram of an example Graphical User Interface (GUI) display with an example hidden font GUI display effect column 700 in accordance with some implementations. The display 700 can display content before the application of a font GUI display effect produced by font formula for average prices greater than \$100 designated as being in italics 702. The content 702 displayed in the GUI display 700 can contain a table with columns for "SYMBOL", "MONTH", and "AVGPRICE." The GUI display can also contain a hidden column 704 that is hidden from being displayed on the GUI display. For example, the hidden column 704 can contain font GUI display effect rows for "AVGPRICE CELL FONT." In the example, a processor can receive a format command to apply a font GUI display effect when AvgPrice is greater than \$100. The font GUI display effect outcome of the format application can be stored in the hidden column 704. For example, the hidden column 704 format font GUI display effect for row (AAPL, JAN, \$99.83) is NULL because \$99.83 is less than \$100 and no font GUI display effect is provided for this range. In the row (AAPL, FEB, \$101.96), the format font GUI display effect stored in the AVGPRICE CELL FONT column is ITALICS because the AVGPRICE for this row is greater than \$100.

[0079] FIG. 7A is a diagram of an example GUI visible content with a cell font GUI display effect applied 450 in accordance with some implementations. For example, applying the font GUI display effect contained in the hidden column 704 from the example in FIG. 7, the font GUI display effect from hidden column 704 can be applied to the adjacent cells to the hidden column 704 in the GUI display 750.

[0080] It will be appreciated that the example NULL font GUI display effect values from the hidden column 704 can have no effect on the corresponding cells in 750 and a default format can be rendered.

[0081] It will be appreciated that data sources can be static or dynamic. Dynamic data sources can have data added, modified, re-indexed, reordered, or deleted. For example, a dynamic table can have one or more rows added, one or more rows modified, or one or more rows deleted. If a dynamic table has one or more rows added, a GUI display effect will be added for those rows in the hidden column 704. If a dynamic table has one or more rows modified, the GUI display effect in the hidden column 704 for those rows can change if the data used to determine the GUI display effect is modified enough to move it beyond the existing GUI display effect threshold in the format command. As the dynamic table data is updated on the GUI display screen, the GUI display effect can also be updated on the GUI display screen.

[0082] Computer instructions for selecting a possible GUI display effect may be hidden. For example, in a database, the instructions may be in a hidden column or columns. The hidden column or columns may be unavailable to a viewer observing a display showing the GUI display effect on visible columns of the same data source. Alternatively, the

hidden column may be available to a user taking a certain action to unhide the column. In that case, the instructions may be configurable by a user of the display. While embodiments are described herein as employing a hidden column, it is understood that multiple hidden columns may be used.

[0083] FIG. 8 is a diagram of an example GUI display with a hidden font column 800 in accordance with some implementations. The GUI display 800 can display content before the application of a font GUI display effect produced by font formula for average prices greater than \$100 designated as requiring italics. The content 802 displayed in the GUI display 800 can contain a table with columns for "SYMBOL", "MONTH", and "AVGPRICE." The GUI display can also contain one or more hidden columns 804 that is hidden from being displayed on the GUI display. For example, the hidden column 804 can contain font GUI display effect rows for "AVGPRICE ROW FONT." In the example, a processor can receive a format command to apply an italics font GUI display effect when AvgPrice is greater than \$100. The font GUI display effect outcome of the format application can be stored in the hidden column 804. For example, the hidden column 804 format font GUI display effect for row (AAPL, JAN, \$99.83) is NULL because \$99.83 is less than \$100 and no font GUI display effect is provided for this value. In the row (AAPL, FEB, \$101.96), the format font GUI display effect stored in the AVGPRICE ROW FONT column is italics because the AVGPRICE for this row is greater than \$100.

[0084] FIG. 8A is a diagram of an example GUI with visible content with a row font GUI display effect applied 850 in accordance with some implementations. For example, applying the font GUI display effect contained in the hidden column 804 from the example in FIG. 8, the font GUI display effect from hidden column 804 can be applied to the corresponding rows in 850.

[0085] It will be appreciated that the example NULL font GUI display effect values from the hidden column 804 can use the default formatting on the corresponding rows in 850.

[0086] It will be appreciated that data sources can be static or dynamic. Dynamic data sources can have data added, modified, re-indexed, reordered, or deleted. For example, a dynamic table can have one or more rows added, one or more rows modified, or one or more rows deleted. If a dynamic table has one or more rows added, a GUI display effect will be added for those rows in the hidden column 804. If a dynamic table has one or more rows modified, the font GUI display effect in the hidden column 804 for those rows can change if the data used to determine the font GUI display effect is modified enough to move it beyond the existing font GUI display effect threshold in the format command. As the dynamic table data is updated on the GUI display screen, the GUI display effect can also be updated on the GUI display screen.

[0087] FIG. 9 is a diagram of an example flow chart for determining and displaying a GUI display effect in accordance with some implementations. Processing begins at 902, when the system receives a format command, for example, t3=t1.formatColumns("Price=(Price>100) ? ITALICS: NULL") to font format cells in the Price column in italics if Price is greater than 100.

[0088] It will be appreciated the GUI display effect is not limited to fonts. The GUI display effect may use different colors, shading, brightness, flashing, motion, sizing, and/or other visual effect to call attention to or differentiate an area

of a display, a browser page or data contained therein. While certain exemplary embodiments are described herein using italics as a GUI display effect, it is understood that any GUI display effect may supplement or substitute for italics as a GUI display effect. Processing continues to **904**.

**[0089]** At **904**, a GUI display effect formula for a formatting column is created. Processing continues to **906**.

**[0090]** At **906**, the GUI display effect formula for a formatting column is hidden from being displayed by the graphical user interface. Processing continues to **908**.

**[0091]** At **908**, the GUI display effect is dynamically computed using the formula in the formula for formatting column.

**[0092]** It will be appreciated that GUI display effects can be pre-computed before determining that a region of a table is displayed. Processing continues to **910**.

**[0093]** At **910**, the GUI display effect for the section of data to be displayed is computed from the hidden GUI display effect formatting column. Processing continues to **912**.

**[0094]** At **912**, the GUI display effect is applied to a section of data to be displayed. Processing continues to **914**.

**[0095]** At **914**, the section of data with the applied GUI display effect is displayed in a graphical user interface.

**[0096]** It will be appreciated that the modules, processes, systems, and sections described above can be implemented in hardware, hardware programmed by software, software instructions stored on a nontransitory computer readable medium or a combination of the above. A system as described above, for example, can include a processor configured to execute a sequence of programmed instructions stored on a nontransitory computer readable medium. For example, the processor can include, but not be limited to, a personal computer or workstation or other such computing system that includes a processor, microprocessor, microcontroller device, or is comprised of control logic including integrated circuits such as, for example, an Application Specific Integrated Circuit (ASIC), a field programmable gate array (FPGA), or graphics processing unit (GPU), or the like. The instructions can be compiled from source code instructions provided in accordance with a programming language such as Java, C, C++, C#.net, assembly or the like. The instructions can also comprise code and data objects provided in accordance with, for example, the Visual Basic™ language, a specialized database query language, or another structured or object-oriented programming language. The sequence of programmed instructions, or programmable logic device configuration software, and data associated therewith can be stored in a nontransitory computer-readable medium such as a computer memory or storage device which may be any suitable memory apparatus, such as, but not limited to ROM, PROM, EEPROM, RAM, flash memory, disk drive and the like.

**[0097]** Furthermore, the modules, processes systems, and sections can be implemented as a single processor or as a distributed processor. Further, it should be appreciated that the steps mentioned above may be performed on a single or distributed processor (single and/or multi-core, or cloud computing system). Also, the processes, system components, modules, and sub-modules described in the various figures of and for embodiments above may be distributed across multiple computers or systems or may be co-located in a single processor or system. Example structural embodi-

ment alternatives suitable for implementing the modules, sections, systems, means, or processes described herein are provided below.

**[0098]** The modules, processors or systems described above can be implemented as a programmed general purpose computer, an electronic device programmed with microcode, a hard-wired analog logic circuit, software stored on a computer-readable medium or signal, an optical computing device, a networked system of electronic and/or optical devices, a special purpose computing device, an integrated circuit device, a semiconductor chip, and/or a software module or object stored on a computer-readable medium or signal, for example.

**[0099]** Embodiments of the method and system (or their sub-components or modules), may be implemented on a general-purpose computer, a special-purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element, an ASIC or other integrated circuit, a digital signal processor, a hardwired electronic or logic circuit such as a discrete element circuit, a programmed logic circuit such as a PLD, PLA, FPGA, PAL, or the like. In general, any processor capable of implementing the functions or steps described herein can be used to implement embodiments of the method, system, or a computer program product (software program stored on a non-transitory computer readable medium).

**[0100]** Furthermore, embodiments of the disclosed method, system, and computer program product (or software instructions stored on a nontransitory computer readable medium) may be readily implemented, fully or partially, in software using, for example, object or object-oriented software development environments that provide portable source code that can be used on a variety of computer platforms. Alternatively, embodiments of the disclosed method, system, and computer program product can be implemented partially or fully in hardware using, for example, standard logic circuits or a VLSI design. Other hardware or software can be used to implement embodiments depending on the speed and/or efficiency requirements of the systems, the particular function, and/or particular software or hardware system, microprocessor, or microcomputer being utilized. Embodiments of the method, system, and computer program product can be implemented in hardware and/or software using any known or later developed systems or structures, devices and/or software by those of ordinary skill in the applicable art from the function description provided herein and with a general basic knowledge of the software engineering and computer networking arts.

**[0101]** Moreover, embodiments of the disclosed method, system, and computer readable media (or computer program product) can be implemented in software executed on a programmed general purpose computer, a special purpose computer, a microprocessor, or the like.

**[0102]** It is, therefore, apparent that there is provided, in accordance with the various embodiments disclosed herein, methods, systems and computer readable media for GUI display effect formatting.

**[0103]** Application Ser. No. 15/154,974, entitled "DATA PARTITIONING AND ORDERING" (Attorney Docket No. W1.1-10057) and filed in the United States Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0104]** Application Ser. No. 15/154,975, entitled “COMPUTER DATA SYSTEM DATA SOURCE REFRESHING USING AN UPDATE PROPAGATION GRAPH” (Attorney Docket No. W1.4-10058) and filed in the United States Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0105]** Application Ser. No. 15/154,979, entitled “COMPUTER DATA SYSTEM POSITION-INDEX MAPPING” (Attorney Docket No. W1.5-10083) and filed in the United States Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0106]** Application Ser. No. 15/154,980, entitled “SYSTEM PERFORMANCE LOGGING OF COMPLEX REMOTE QUERY PROCESSOR QUERY OPERATIONS” (Attorney Docket No. W1.6-10074) and filed in the United States Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0107]** Application Ser. No. 15/154,983, entitled “DISTRIBUTED AND OPTIMIZED GARBAGE COLLECTION OF REMOTE AND EXPORTED TABLE HANDLE LINKS TO UPDATE PROPAGATION GRAPH NODES” (Attorney Docket No. W1.8-10085) and filed in the United States Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0108]** Application Ser. No. 15/154,984, entitled “COMPUTER DATA SYSTEM CURRENT ROW POSITION QUERY LANGUAGE CONSTRUCT AND ARRAY PROCESSING QUERY LANGUAGE CONSTRUCTS” (Attorney Docket No. W2.1-10060) and filed in the United States Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0109]** Application Ser. No. 15/154,985, entitled “PARSING AND COMPILING DATA SYSTEM QUERIES” (Attorney Docket No. W2.2-10062) and filed in the United States Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0110]** Application Ser. No. 15/154,987, entitled “DYNAMIC FILTER PROCESSING” (Attorney Docket No. W2.4-10075) and filed in the United States Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0111]** Application Ser. No. 15/154,988, entitled “DYNAMIC JOIN PROCESSING USING REAL-TIME MERGED NOTIFICATION LISTENER” (Attorney Docket No. W2.6-10076) and filed in the United States Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0112]** Application Ser. No. 15/154,990, entitled “DYNAMIC TABLE INDEX MAPPING” (Attorney Docket No. W2.7-10077) and filed in the United States Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0113]** Application Ser. No. 15/154,991, entitled “QUERY TASK PROCESSING BASED ON MEMORY ALLOCATION AND PERFORMANCE CRITERIA” (Attorney Docket No. W2.8-10094) and filed in the United States

Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0114]** Application Ser. No. 15/154,993, entitled “A MEMORY-EFFICIENT COMPUTER SYSTEM FOR DYNAMIC UPDATING OF JOIN PROCESSING” (Attorney Docket No. W2.9-10107) and filed in the United States Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0115]** Application Ser. No. 15/154,995, entitled “QUERY DISPATCH AND EXECUTION ARCHITECTURE” (Attorney Docket No. W3.1-10061) and filed in the United States Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0116]** Application Ser. No. 15/154,996, entitled “COMPUTER DATA DISTRIBUTION ARCHITECTURE” (Attorney Docket No. W3.2-10087) and filed in the United States Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0117]** Application Ser. No. 15/154,997, entitled “DYNAMIC UPDATING OF QUERY RESULT DISPLAYS” (Attorney Docket No. W3.3-10059) and filed in the United States Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0118]** Application Ser. No. 15/154,998, entitled “DYNAMIC CODE LOADING” (Attorney Docket No. W3.4-10065) and filed in the United States Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0119]** Application Ser. No. 15/154,999, entitled “IMPORTATION, PRESENTATION, AND PERSISTENT STORAGE OF DATA” (Attorney Docket No. W3.5-10088) and filed in the United States Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0120]** Application Ser. No. 15/155,001, entitled “COMPUTER DATA DISTRIBUTION ARCHITECTURE” (Attorney Docket No. W3.7-10079) and filed in the United States Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0121]** Application Ser. No. 15/155,005, entitled “PERSISTENT QUERY DISPATCH AND EXECUTION ARCHITECTURE” (Attorney Docket No. W4.2-10089) and filed in the United States Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0122]** Application Ser. No. 15/155,006, entitled “SINGLE INPUT GRAPHICAL USER INTERFACE CONTROL ELEMENT AND METHOD” (Attorney Docket No. W4.3-10063) and filed in the United States Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0123]** Application Ser. No. 15/155,007, entitled “GRAPHICAL USER INTERFACE DISPLAY EFFECTS FOR A COMPUTER DISPLAY SCREEN” (Attorney Docket No. W4.4-10090) and filed in the United States Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0124]** Application Ser. No. 15/155,009, entitled “COMPUTER ASSISTED COMPLETION OF HYPERLINK COMMAND SEGMENTS” (Attorney Docket No. W4.5-10091) and filed in the United States Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0125]** Application Ser. No. 15/155,010, entitled “HISTORICAL DATA REPLAY UTILIZING A COMPUTER SYSTEM” (Attorney Docket No. W5.1-10080) and filed in the United States Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0126]** Application Ser. No. 15/155,011, entitled “DATA STORE ACCESS PERMISSION SYSTEM WITH INTERLEAVED APPLICATION OF DEFERRED ACCESS CONTROL FILTERS” (Attorney Docket No. W6.1-10081) and filed in the United States Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0127]** Application Ser. No. 15/155,012, entitled “REMOTE DATA OBJECT PUBLISHING/SUBSCRIBING SYSTEM HAVING A MULTICAST KEY-VALUE PROTOCOL” (Attorney Docket No. W7.2-10064) and filed in the United States Patent and Trademark Office on May 14, 2016, is hereby incorporated by reference herein in its entirety as if fully set forth herein.

**[0128]** While the disclosed subject matter has been described in conjunction with a number of embodiments, it is evident that many alternatives, modifications and variations would be, or are, apparent to those of ordinary skill in the applicable arts. Accordingly, Applicants intend to embrace all such alternatives, modifications, equivalents and variations that are within the spirit and scope of the disclosed subject matter.

1. A system for formatting data for display on a computer display screen, the system comprising:

- one or more hardware processors;
- a computer display screen configured to display data via a graphical user interface (GUI);
- a computer readable data storage device coupled to the one or more hardware processors, the computer readable data storage device having stored thereon software instructions that, when executed by the one or more hardware processors, cause the one or more hardware processors to perform operations including:
  - creating a GUI display effect formula for a formatting column;
  - hiding the formatting column from being displayed by the graphical user interface on the computer display screen;
  - dynamically computing a GUI display effect using the GUI display effect formula for a formatting column;
  - choosing a section of data to be displayed on the computer display screen via the graphical user interface;
  - computing the GUI display effect for the section of data to be displayed on the computer display screen via the graphical interface using the hidden formatting column;
  - applying the computed GUI display effect to the section of data to be displayed;
  - displaying on the computer display screen the section of data with the applied computed GUI display effect.

2. The system of claim 1, wherein the operations further comprise applying the GUI display effect formatting by applying a formula to associated data in a same row of data.

3. The system of claim 1, wherein applying the GUI display effect formatting includes applying a GUI display effect to at least one of an entire row of data, an entire column of data, an entire table, a tab, a browser, and a graphical user interface.

4. The system of claim 1, wherein the section of data to be displayed on the graphical interface using the hidden formatting column is determined by scrolling.

5. The system of claim 1, wherein the operations further comprise:

- receiving updates to the section of data to be displayed on the graphical user interface;
- using a processor, dynamically computing a GUI display effect for the updates to the section of data using the GUI display effect formula for a formatting column;
- choosing a section of data to be displayed on the graphical user interface;
- computing the GUI display effect for the section of data to be displayed on the graphical interface using the hidden formatting column;
- applying the computed GUI display effect to the section of data to be displayed;
- displaying the section of data with the applied computed GUI display effect.

6. The system of claim 1, wherein computing the GUI display effect occurs prior to displaying the section of data.

7. The system of claim 1, wherein computing the GUI display effect occurs at the time of displaying the section of data.

8. A method for formatting data displayed in a graphical user interface (GUI), the method comprising:

- creating a GUI display effect formula for a formatting column;
- hiding the formatting column from being displayed by the graphical user interface;
- using a processor, dynamically computing a GUI display effect using the GUI display effect formula for a formatting column;
- choosing a section of data to be displayed on the graphical user interface;
- computing the GUI display effect for the section of data to be displayed on the graphical interface using the hidden formatting column;
- applying the computed GUI display effect to the section of data to be displayed;
- displaying the section of data with the applied computed GUI display effect.

9. The method of claim 8, further comprising applying the GUI display effect formatting by applying a formula to associated data in a same row of data.

10. The method of claim 8, wherein applying the GUI display effect formatting includes applying a GUI display effect to at least one of an entire row of data, an entire column of data, an entire table, a tab, a browser, and a graphical user interface.

11. The method of claim 8, wherein the section of data to be displayed on the graphical interface using the hidden formatting column is determined by scrolling.

- 12. The method of claim 8, further comprising:
  - receiving updates to the section of data to be displayed on the graphical user interface;
  - using a processor, dynamically computing a GUI display effect for the updates to the section of data using the GUI display effect formula for a formatting column;

choosing a section of data to be displayed on the graphical user interface;

computing the GUI display effect for the section of data to be displayed on the graphical interface using the hidden formatting column;

applying the computed GUI display effect to the section of data to be displayed;

displaying the section of data with the applied computed GUI display effect.

**13.** The method of claim **8**, wherein computing the GUI display effect occurs prior to displaying the section of data.

**14.** A nontransitory computer readable medium having stored thereon software instructions that, when executed by one or more processors, cause the one or more processors to perform operations including:

creating a GUI display effect formula for a formatting column;

hiding the formatting column from being displayed by the graphical user interface on the computer display screen;

dynamically computing a GUI display effect using the GUI display effect formula for a formatting column;

choosing a section of data to be displayed on the computer display screen via the graphical user interface;

computing the GUI display effect for the section of data to be displayed on the computer display screen via the graphical interface using the hidden formatting column;

applying the computed GUI display effect to the section of data to be displayed;

displaying on the computer display screen the section of data with the applied computed GUI display effect.

**15.** The nontransitory computer readable medium of claim **14**, wherein the operations further comprise applying

the GUI display effect formatting by applying a formula to associated data in a same row of data.

**16.** The nontransitory computer readable medium of claim **14**, wherein applying the GUI display effect formatting includes applying a GUI display effect to at least one of an entire row of data, an entire column of data, an entire table, a tab, a browser, and a graphical user interface.

**17.** The nontransitory computer readable medium of claim **14**, wherein the section of data to be displayed on the graphical interface using the hidden formatting column is determined by scrolling.

**18.** The nontransitory computer readable medium of claim **14**, wherein the operations further comprise:

receiving updates to the section of data to be displayed on the graphical user interface;

using a processor, dynamically computing a GUI display effect for the updates to the section of data using the GUI display effect formula for a formatting column;

choosing a section of data to be displayed on the graphical user interface;

computing the GUI display effect for the section of data to be displayed on the graphical interface using the hidden formatting column;

applying the computed GUI display effect to the section of data to be displayed;

displaying the section of data with the applied computed GUI display effect.

**19.** The nontransitory computer readable medium of claim **14**, wherein computing the GUI display effect occurs prior to displaying the section of data.

**20.** The nontransitory computer readable medium of claim **14**, wherein computing the GUI display effect occurs at the time of displaying the section of data.

\* \* \* \* \*