

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
15 March 2007 (15.03.2007)

PCT

(10) International Publication Number
WO 2007/030696 A2

(51) International Patent Classification: Not classified

(21) International Application Number:
PCT/US2006/034993

(22) International Filing Date:
7 September 2006 (07.09.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/714,622 7 September 2005 (07.09.2005) US
11/401,470 10 April 2006 (10.04.2006) US

(71) Applicant (for all designated States except US): MICROSOFT CORPORATION [US/US]; One Microsoft Way, Redmond, WA 98052-6399 (US).

(72) Inventors: MCCORMACK, Michael, J.; One Microsoft Way, Redmond, WA 98052-6399 (US). KENNEY, Adam, Russell; One Microsoft Way, Redmond, WA 98052-6399 (US). BRYANT, Christopher, W.; One Microsoft Way, Redmond, WA 98052-6399 (US). COVINGTON, Clinton, Dee; One Microsoft Way, Redmond, WA 98052-6399 (US). DICKINSON, Richard, L.; One Microsoft Way, Redmond, WA 98052-6399 (US). COGGINS, Robert, E.; One Microsoft Way, Redmond, WA 98052-6399 (US). KROUT, Kelly, Michael; One Microsoft way, Redmond, WA 98052-6399 (US). SELCA, Victore; One Microsoft Way, Redmond, WA 98052-6399 (US). WOODALL, Zachary, P.; One Microsoft Way, Redmond, WA 98052-6399 (US). HOUGH, Dan, Jacob; One Microsoft Way, Redmond, WA 98052-6399 (US).

(74) Agent: MICROSOFT CORPORATION; Attn.: Sharon Rydberg, (sharonr-21/2029), LCA, International Patent Department, One Microsoft Way, 21/2029, Redmond, WA 98052-6399 (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, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

Declarations under Rule 4.17:

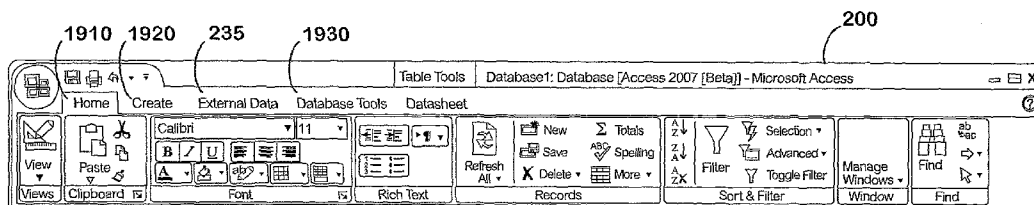
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))
- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))

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.

(54) Title: COMMAND USER INTERFACE FOR DISPLAYING SELECTABLE FUNCTIONALITY CONTROLS IN A DATABASE APPLICATION



(57) Abstract: An improved user interface is provided for displaying selectable software functionality controls of a database application. When a task-based functionality tab is selected from the user interface, selectable functionality controls provided by the associated database application for performing aspects of a task related to the selected tab are presented in a ribbon-shaped user interface above a workspace in which a user is entering or editing data. Upon selection of a different task-based functionality tab, the presentation of functionality controls associated with the first task is removed, and selectable functionality controls associated with the second selected task-based functionality tab are presented in the user interface. The selectable functionality controls presented in the user interface are grouped into logical groupings for more efficient utilization. An extensibility model is provided for allowing users to develop and use customized user interfaces and to allow utilization of legacy user interfaces.

WO 2007/030696 A2

**COMMAND USER INTERFACE FOR DISPLAYING SELECTABLE
FUNCTIONALITY CONTROLS IN A DATABASE APPLICATION****Background**

With the advent of the computer age, computer and software users have grown accustomed to user-friendly software applications that help them write, calculate, organize, prepare presentations, send and receive electronic mail, store and manipulate data, make music, and the like. For example, modern database applications allow users to store, organize and manipulate data. Modern electronic word processing applications allow users to prepare a variety of useful documents. Modern spreadsheet applications allow users to enter, manipulate, and organize data. Modern electronic slide presentation applications allow users to create a variety of slide presentations containing text, pictures, data or other useful objects.

To assist users to locate and utilize functionality of a given software application, for example, a database application, a user interface containing a plurality of generic functionality controls is typically provided along an upper, lower or side edge of a displayed workspace in which the user may enter, copy, manipulate and format text or data. Such functionality controls often include selectable buttons with such names as "file," "edit," "view," "insert," "format," and the like. Typically, selection of one of these top-level functionality buttons, for example "insert," causes a drop-down menu to be deployed to expose one or more selectable functionality controls associated with the top-level functionality, for example, "table" under a top-level functionality of "insert."

After a user selects a desired functionality control, or if the user moves the mouse cursor to a different location, the drop-down menu typically disappears. If the user determines that a functionality of the first drop-down menu was the desired functionality, the user must remember which top-level functionality was selected, reselect that functionality and then find the desired functionality control all over again. Accordingly, in order to use the functionality of a given software application, the user must know the desired functionality is available under one of the selectable buttons, or the user must select different top-level functionalities until the desired specific functionality is located in one of many deployed menus. Such a method of searching for desired functionality is cumbersome and time-consuming, particularly for less experienced users, and when new functionality is added by developers of the software application, the new functionality may never be utilized unless the user is somehow educated as to its existence.

In addition, often a given user desires to customize a user interface provided by a given software application, or a given user often desires to utilize a legacy user interface of a previous version of a software application with which one or more data objects or documents has been developed or edited. If the user is not able to customize the user interface, the utilization experience may be greatly diminished. If the user is not able to utilize legacy user interfaces, the user may be required to expend significant resources updating data objects or documents to comply with later versions of the software application.

It is with respect to these and other considerations that the present invention has been made.

Summary

This summary is provided to introduce a selection of concepts in a simplified form that are further described below in the detailed description. This summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended as an aid in determining the scope of the claimed subject matter.

Embodiments of the present invention solve the above and other problems by providing an improved user interface for displaying selectable software functionality controls of a database application and for presenting logical groupings of particular functionality controls associated with a selected top-level functionality. Generally, aspects of the present invention provide for organization of the functionality of a database application into task-based modes. The modes are associated with tabs in a ribbon-shaped user interface, and the tabs are labeled with descriptive text associated with different functionality modes or tasks. Underneath a row of top-level functionality tabs, functionalities associated with a given top-level functionality tab are presented in logical groupings. Selection of a particular tab switches modes of the user interface to present controls for functionalities associated with the selected tab.

According to one embodiment of the invention, methods and systems for providing functionality from a software application via an improved user interface are provided. A plurality of functionalities available from a database application is organized according to one or more tasks that may be performed with the database application. A user interface tab for each of the one or more tasks is provided in the user interface. Upon receiving an indication of a selection of a given user interface tab, one or more selectable functionality controls are provided in the user interface for selecting one or more functionalities organized under a given task associated with the selected user interface tab.

According to another embodiment of the invention, upon receiving a selection of a given data object, for example, a database report, the one or more selectable functionality controls is/are dynamically replaced in the user interface with one or more selectable functionality controls that are relevant to the selected object so that the user interface becomes contextually relevant to the selected object.

According to another embodiment, an application programming interface (API) method may be utilized for submitting customizations of the user interface. In addition, a property or condition may be set on the user interface to allow utilization of a legacy or previous version of a user interface so that backward compatibility is provided with data objects or documents prepared using legacy or previous versions of an associated software application.

These and other features and advantages will be apparent from a reading of the following detailed description and a review of the associated drawings. It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the invention as claimed.

Brief Description of the Drawings

Fig. 1 is a block diagram showing the architecture of a personal computer that provides an illustrative operating environment for embodiments of the present invention.

Fig. 2 is an illustration of a computer screen display showing a ribbon-shaped user interface for displaying task-based top-level functionality tabs and for displaying a plurality of functionalities available under a selected file menu control.

Fig. 3 is an illustration of a computer screen display showing a ribbon-shaped user interface for displaying task-based top-level functionality tabs and showing a selected Data tab for providing commands that are common across a number of data objects in a database application.

Fig. 4 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing a selected Insert tab for providing commands for new objects into a database file.

Fig. 5 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing a selected Page Layout tab for providing commands for affecting how a given data in a particular object is displayed.

Fig. 6 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing a selected External Data tab for providing commands for importing data from a source outside an in-use database.

Fig. 7 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing a selected Advanced Tools tab is for providing commands for performing maintenance on a given database.

Fig. 8A illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing a selected Source Control tab for providing commands for utilizing functionality of a third party add in application.

Fig. 8B is an illustration of a computer screen display showing a ribbon-shaped user interface for displaying task-based top-level functionality tabs and showing a Print Preview tab for providing commands for affecting print settings for a selected database object.

Fig. 9 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing contextual tabs for providing commands for formatting and layout for database forms.

Fig. 10 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing contextual tabs for providing commands for formatting and layout for database reports.

Fig. 11 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing contextual tabs for providing commands for database form and database report layout.

Fig. 12 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing a contextual tab for designing a database form in a database design mode.

Fig. 13 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing a contextual tab for providing commands for building a database query in a database design view.

Fig. 14 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing a contextual tab for providing commands for building a macro in a database design view.

Fig. 15 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing a contextual tab for building a relationship between tables or queries in a database design view.

Fig. 16 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing a contextual tab for providing commands for building a pivot table in a database design view.

Fig. 17 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing a contextual tab for providing commands for building a pivot chart in a database design view.

Fig. 18 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing a contextual tab for providing commands for building a Table in a database design view.

Fig. 19 is ribbon-shaped user interface for displaying task-based top-level functionality tabs and showing a selected Home tab for providing commands that are common across a number of data objects in a database application.

Fig. 20 is an illustration of a computer screen display showing the user interface illustrated in Fig. 19 and showing a selected Create tab for providing commands for creating and/or inserting a database object in a database file.

Fig. 21 illustrates a computer screen display showing the user interface illustrated in Fig. 19 and showing a selected External Data tab.

Fig. 22 illustrates a computer screen display showing the user interface illustrated in Fig. 19 and showing a selected Database Tools tab for providing commands for performing maintenance on a given database.

Fig. 23 illustrates a computer screen display showing the user interface illustrated in Fig. 19 and showing a contextual tab for providing database functionality available for application to a table object.

Fig. 24 illustrates a computer screen display showing a ribbon-shaped user interface for displaying task-based top-level functionality tabs and showing a Print Preview tab for providing commands for affecting printing settings for a selected database object.

Detailed Description

As briefly described above, embodiments of the present invention are directed to an improved user interface for displaying selectable software functionality controls of a database application. As will be described in detail below, when one task-based functionality tab is selected from the user interface of the present invention, selectable functionality controls provided by the associated database application for performing aspects of a task related to the selected task-based functionality tab are presented in a ribbon-shaped user interface above a

workspace in which the user is entering or editing data. Upon selection of a different task-based functionality tab, the presentation of functionality controls associated with the first task is removed, and selectable functionality controls associated with the second selected task-based functionality tab are presented in the user interface. The selectable functionality controls presented in the user interface are grouped into logical groupings for more efficient utilization. An extensibility model is provided for allowing users to develop and use customized user interfaces and to allow utilization of legacy user interfaces.

In the following detailed description, references are made to the accompanying drawings that form a part hereof, and in which are shown by way of illustrations specific embodiments or examples. These embodiments may be combined, other embodiments may be utilized, and structural changes may be made without departing from the spirit or scope of the present invention. The following detailed description is therefore not to be taken in a limiting sense and the scope of the present invention is defined by the appended claims and their equivalents.

Referring now to the drawings, in which like numerals refer to like elements through the several figures, aspects of the present invention and an exemplary computing operating environment will be described. Figure 1 and the following discussion are intended to provide a brief, general description of a suitable computing environment in which the invention may be implemented. While the invention will be described in the general context of program modules that execute in conjunction with an application program that runs on an operating system on a personal computer, those skilled in the art will recognize that the invention may also be implemented in combination with other program modules.

Generally, program modules include routines, programs, components, data structures, and other types of structures that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

Embodiments of the invention may be implemented as a computer process (method), a computing system, or as an article of manufacture, such as a computer program product or computer readable media. The computer program product may be a computer storage media

readable by a computer system and encoding a computer program of instructions for executing a computer process. The computer program product may also be a propagated signal on a carrier readable by a computing system and encoding a computer program of instructions for executing a computer process.

With reference to Figure 1, one exemplary system for implementing the invention includes a computing device, such as computing device 100. In a basic configuration, the computing device 100 typically includes at least one processing unit 102 and system memory 104. Depending on the exact configuration and type of computing device, the system memory 104 may be volatile (such as RAM), non-volatile (such as ROM, flash memory, etc.) or some combination of the two. System memory 104 typically includes an operating system 105 suitable for controlling the operation of a networked personal computer, such as the WINDOWS® operating systems from MICROSOFT CORPORATION of Redmond, Washington. The system memory 104 may also include one or more software applications 106 and may include program data 107. This basic configuration is illustrated in Figure 1 by those components within dashed line 108.

According to embodiments of the invention, the application 106 may comprise many types of software applications, such as database applications, word processing applications, spreadsheet applications, slide presentation applications, and the like. The application 106 may also include a multiple-functionality software application for providing many types of functionalities. Such a multiple-functionality application may include a number of program modules, such as a word processing program, a spreadsheet program, a slide presentation program, a database program, and the like. An example of such a multiple-functionality application is OFFICE™ manufactured by MICROSOFT CORPORATION. According to embodiments of the present invention, the application 106 is illustrative of a database application with which data may be stored, maintained manipulated and processed. An example of such a database application 106 is ACCESS® manufactured by MICROSOFT CORPORATION.

The computing device 100 may have additional features or functionality. For example, the computing device 100 may also include additional data storage devices (removable and/or non-removable) such as, for example, magnetic disks, optical disks, or tape. Such additional storage is illustrated in Figure 1 by removable storage 109 and non-removable storage 110. Computer storage media may include volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage

of information, such as computer readable instructions, data structures, program modules, or other data. System memory 104, removable storage 109 and non-removable storage 110 are all examples of computer storage media. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computing device 100. Any such computer storage media may be part of device 100. Computing device 100 may also have input device(s) 112 such as keyboard, mouse, pen, voice input device, touch input device, etc. Output device(s) 114 such as a display, speakers, printer, etc. may also be included. These devices are well known in the art and need not be discussed at length here.

The computing device 100 may also contain communication connections 116 that allow the device to communicate with other computing devices 118, such as over a network in a distributed computing environment, for example, an intranet or the Internet. Communication connection 116 is one example of communication media. Communication media may typically be embodied by computer readable instructions, data structures, program modules, or other data in a modulated data signal, such as a carrier wave or other transport mechanism, and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. The term computer readable media as used herein includes both storage media and communication media.

Fig. 2 is an illustration of a computer screen display showing a ribbon-shaped user interface of a database application 106 for displaying task-based top-level functionality tabs and for displaying a plurality of functionalities available under a selected top-level functionality tab. As will be described in detail herein, the display of available functionality controls in the user interface of the present invention is dynamically changed based on a selected mode of operation or task of the database application or based on a context of a selected data object. As should be appreciated, the user interface 200 modes of operation and layouts described below are for purposes of example only and are not limiting of the additional modes of operation and additional displays of logical groupings of functionality controls that may be provided in accordance with embodiments of the present invention.

According to one embodiment, a first selectable control 210 is displayed consistently across all modes of the user interface 200 for providing a plurality of standard application functions such as the “New” file control 242, the “Open” file control 245, the “Save” control, the “Print” control, and the like. That is, selection of the “File” control 210 causes a display of the selectable controls 242, 245, 260 as illustrated in Fig. 2. Adjacent to the “File” control 210 is a quick action toolbar 215 for providing one or more standard functionalities such as save, print, undo, help, and the like for quick selection by a user. As will be illustrated in Figs. 2 - 18, the quick action toolbar 215 may be located at different locations on the user interface, for example, top, bottom, top right, bottom right, etc. As should be appreciated, selection of one of the functionality controls from the quick action toolbar 215 allows for the quick execution of the desired functionality without the need for deploying a secondary functionality menu or without selecting a different top-level functionality control 220, 225, 230, 235, 240, described herein.

According to embodiments of the present invention, selection of one of the top-level task-based functionality tabs 220, 225, 230, 235, 240 (illustrated in Fig. 3) changes a display mode of the user interface 200 such that the user interface 200 dynamically displays selectable controls for one or more functionalities of the associated database application 106 such that the displayed functionality controls are associated with the selected tab and associated functionality mode. For example, as described below, selection of the “Data” tab 220 places the user interface 200 in a “data” mode, and selectable functionality controls for providing data functions of the database application 106 are displayed immediately beneath the row of top-level task-based tabs. Similarly, selection of the “Insert” tab 225 causes a display of one or more selectable functionality controls for providing insert functionalities of the database application 106 immediately beneath the row of task-based tabs. Selection of the “Page Layout” tab 230 causes a display of one or more functionality controls for providing page layout functionality of the database application 106, and so on as described herein. As should be appreciated, the task-based tabs 210-235, illustrated in Fig. 2, are for purposes of illustration only and are not restrictive of the numerous additional task-based tabs that may be displayed in the user interface 200 for causing a display of selectable functionality controls for a given task or mode enabled by the associated database application 106.

According to an embodiment, if a focus is temporarily received on a second top-level task-based tab, for example, the “Insert” tab, described below, the user interface 200 may be dynamically changed to temporarily show the logical groupings of functionality controls that

will be populated in the user interface if the focused-on tab is actually selected. Upon cessation of the focus (e.g., mouse-over focus) on the second user interface tab, the logical groupings of functionality controls previously displayed are re-displayed in the user interface. Thus, a “preview” of controls associated with a given tab or mode of operation may be obtained. According to one embodiment, if during the “preview” deployment of the second set of controls a given control is selected, the functionality of the selected control is applied to a selected data object. Once application of the selected functionality is done, the “preview” deployment is dismissed and a display of the originally deployed controls is returned unless a selection of the second task-based tab is received to persist the display of the second set of controls in the user interface. Alternatively, selection of one of the “temporarily” displayed controls may cause a persistent display of the second set of controls as if the second task-based tab was selected.

Fig. 3 is an illustration of a computer screen display showing a ribbon-shaped user interface for displaying task-based top-level functionality tabs and showing a selected Data tab for providing commands that are common across a number of data objects in a database application. According to an embodiment these commands are advantageous for working with (e.g., manipulating) data in a table and/or for committing data to a table. Upon selection of the “Data” tab 220, a selectable functionality control section immediately underneath the row of task-based tabs 210-240 is dynamically deployed for displaying selectable functionality controls associated with the selected tab 220. According to embodiments of the present invention, when a particular tab, for example, the “Data” tab 220 is selected, functionality available from the associated database application for performing the selected task, or associated with the selected mode of operation, for example, data operations, is displayed beneath the row of task-based tabs in one or more logical groupings of functionality controls. The functionality controls displayed in the one or more logical groupings of controls represent a subset of the overall functionality of the associated database application 106 where the subset is associated with a mode of operation of the database application associated with the selected tab, for example, a data mode, insert mode, or page layout mode.

Referring to Fig. 3, a first logical grouping of controls 310 is displayed under a heading “Clipboard.” According to embodiments of the present invention, the clipboard section 310 may include such selectable controls as a “Cut,” “Copy,” “Paste,” “Select All,” etc. for providing clipboard-type functionalities for use on one or more data objects. Adjacent to the clipboard section 310, a second logical grouping 315 is presented under the

heading "Font" for providing font functionalities for use with one or more data objects. A third logical grouping 320 is provided under the heading "Font Alignment" for providing selectable functionality controls for applying font alignment properties to one or more data objects. A next logical grouping of functionality controls 330 is displayed under the heading "Data Formatting" for providing one or more selectable controls for applying data formatting properties to one or more selected objects. Additional logical groupings of functionality controls 335, 340, 345 and 350 provide groupings of selectable functionality controls for data records, data editing, data sorting and filtering, and data find functionality, respectively. As should be appreciated, the logical groupings of functionality controls illustrated in Fig. 3 are for purposes of illustration only and are not limiting of additional functionality controls and different arrangements of functionality controls that may be provided in logical groupings under the row of task-based tabs 210-240 upon selection of a given task-based tab.

According to one embodiment, once a top-level task-based tab 210-240 is selected, and the associated logical groupings of functionality controls are deployed, as illustrated in Fig. 3, the deployed logical groupings of functionality controls remain deployed in the user interface 200 until the user interface 200 is placed into a different operating mode associated with a different top-level task of the database application 106 by the selection of a different task-based tab 210 - 240 or until the context of the user interface is changed relative to a context of one or more selected data objects, as described below with reference to Figs. 9 - 18.

Fig. 4 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing a selected Insert tab for providing commands for new objects into a database file. According to an embodiment, these commands may be used for inserting new tables, forms, reports, queries, macros and modules into a database. Upon selection of the "Insert" tab 225, a different set of logical groupings of functionality controls is deployed for providing access to functionalities of the associated database application 106 associated with an insert task or mode of operation. As illustrated in Fig. 4, a first logical grouping of controls 410 is provided under the heading "Tables and Lists." As should be appreciated, the selectable controls displayed under the "Tables and Lists" logical grouping 410 allow for selectively inserting data tables and data lists associated with one or more data objects. A second logical grouping 415 is deployed under the heading "Forms" and provides one or more functionality controls for inserting one of different types of forms into a given database data object. Other logical groupings of functionality controls under the "Insert" tab 225 include the "Reports" grouping 420, the "Other" grouping 425, and the "Field and Columns"

grouping 430. As illustrated in the lower portion of Fig. 4, upon selection of a given functionality control in one or more of the logical groupings of controls, one or more galleries of additional functionality controls 435, 440, 445, 450 may be provided for displaying additional controls for inserting one or more types of data structuring objects, such as tables, lists, forms, reports, and the like.

Fig. 5 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing a selected Page Layout tab for providing commands for affecting how a given data in a particular object is displayed. Upon selection of the "Page Layout" tab 230, one or more logical groupings of functionality controls is deployed underneath the row of task-based functionality tabs 210-240. A first logical grouping of functionality controls 510 is deployed under the heading "Page" and provides selectable controls for applying page layout functionality of the associated database application 106 to one or more data objects available to the database application 106. For example, controls are provided for setting page margins, setting page display properties, and setting page display sizes. Beneath the logical grouping of controls 510 is deployed a gallery of controls 520 for providing one or more preformatted page layout properties to a given database data object. For example, a user may select one of the illustrated page layout formats from the gallery of controls 520 for having the associated layout properties automatically applied to a given data object. For example, if the "Normal" control is selected, the properties associated with the "Normal" layout, including the margin widths on the top, bottom, left and/or right, automatically will be applied to a selected data object.

As described above with reference to Fig. 3, once a given task-based tab 210-240 is selected, and an associated one or more logical groupings of functionality controls is/are deployed, the deployed one or more logical groupings of functionality controls remains deployed until the mode of operation of the user interface 200 is changed by selection of a different task-based tab 210-240. As should be appreciated, a default operating mode, for example, "data" mode, may be automatically deployed along with its associated logical groupings of functionality controls upon launching an instance of the associated database application 106.

Fig. 6 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing a selected External Data tab for providing commands for importing data from a source outside an in-use database. Selection of the "External Data" tab 235 places the user interface 200 in a mode for providing functionality of the associated database application 106 in association with external data. Upon selection of the "External Data" tab 235, a

plurality of logical groupings of functionality controls is deployed underneath the row of task-based tabs 210-240 that are associated with a task or mode of operation related to external data. A first logical grouping of controls 610 is provided under the heading "Get External Data" and provides one or more functionality controls for causing the associated database application 106 to retrieve external data from a number of sources, including saved data, data maintained in a shared database, data maintained or enabled from a different software application, and the like.

Referring still to Fig. 6, a second logical grouping of functionality controls 620 is deployed under the heading "Export Data" and provides one or more functionality controls for causing the associated database application 106 to export data from the database application 106 to an external data source or data site. A third logical grouping of functionality controls 625 is deployed under the heading "Collect and Update" and includes one or more functionality controls for handling external data via electronic mail transmission and management. A next logical grouping of functionality controls 630 is deployed under the heading "Offline Sharepoint Lists" and provides one or more functionality controls for processing data in an offline environment and for synchronizing data processed in an offline environment back to a server-based database once an online connection between a client database application and a server-based database application is reestablished.

Underneath the logical groupings of functionality controls are additional functionality controls that may be provided under the logical groupings of functionality controls where additional controls are available or desired. For example, the dropdown 645 provides additional functionality controls for processing data in an offline environment. The dropdowns 635 and 645 contain additional functionality controls for obtaining external data and for exporting data. As should be appreciated, the dropdowns 635, 640, 645 may be deployed anytime the logical groupings 610, 620, 625, 630 do not include all available functionality controls under each of the respective logical grouping headings. As should be appreciated, the logical groupings and associated functionality controls illustrated in Fig. 6 are for purposes of example only and are not limiting of the number of different functionality controls and logical groupings of functionality controls that may be deployed under the "External Data" tab 235.

Fig. 7 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing a selected Advanced Tools tab is for providing commands for performing maintenance on a given database. Upon selection of the "Advanced Tools" tab 240, a number of logical groupings of functionality controls 710, 720, 730, 740 are deployed for

providing functionality controls for obtaining advanced features, such as database analysis, upsize functions and additional database tools, including database security functions. For example, the logical grouping of functionality controls 710 deployed under the heading "Analyze" provides advanced data analysis tools. The logical grouping 720 under the heading "Upsize To" provides one or more advanced tools for upsizing data to a variety of different database systems. The logical grouping 730 under the heading "Database Tools" provides a number of advanced tools for operating on the database application 106 and associated data. The dropdown or deployed menus 750, 760 illustrate additional functionality controls not deployed in the logical groupings, including additional security functions illustrated in the menu 750 and data synchronization functions illustrated in the menu 760.

Fig. 8A illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing a selected Source Control tab for providing commands for utilizing functionality of a third party add-in application. According to one embodiment, the Source Control tab may be provided by a third party add-in, such as Visual SourceSafe (VSS), and is only surfaced when add-in application is installed on the computer 100. Upon a selection of the "Source Control" tab 810, a variety of functionality controls are deployed in one or more logical groupings of functionality controls 820, 830, 840. For example, a first logical grouping of controls 820 is deployed under the heading "Work with Source Control" and provides functionalities, particularly useful to database application developers, for designing particular implementations of data input and output from a client application to a data source. A second logical grouping of functionality controls 830 is deployed under the heading "Show/Hide" and provides one or more functionality controls for setting display properties on one or more data objects. A third logical grouping of controls 840 is deployed under the heading "Manage" and provides one or more functionality controls for managing the operation of, relation between, and communication between various data objects.

Fig. 8B is an illustration of a computer screen display showing a ribbon-shaped user interface for displaying task-based top-level functionality tabs and showing a Print Preview tab for providing commands for affecting printing settings for a selected database object. As illustrated in Fig 8B, a number of logical groupings of controls, 850, 860, 865, 870 are provided for choosing different print settings for an associated database object. For example, the logical grouping 865 contains controls for selectively printing a database object in "Portrait" orientation or "Landscape" orientation.

The user interface modes of operation described above with reference to Figs. 2 - 8 provide particular deployments of functionality controls in association with particular modes

of operation, for example, insert, page layout, external data, advanced tools, etc. According to embodiments of the present invention, the user interface 200 may be populated with additional logical groupings of functionality controls based on the context of a given selected data object so that the user interface 200 provides logical groupings of functionality controls that immerse the user with functionality responsive to a particular data object. For example, if the user is in a particular mode of operation, for example, upon selection of the "Data" tab 220, and the user selects a particular data object, for example, a table object, the logical groupings of functionality controls deployed in the user interface 200 may be dynamically changed so that functionality controls particular to features of the associated database application for operating on the selected table object are deployed in the user interface 200. Thus, typical features required for processing the selected table object are automatically deployed in the user interface 200 for use by the user. If the selection of a given database object is terminated, or if the mouse cursor or other electronic pointer is focused on an area of the user interface not associated with a particular database object, the previously deployed user interface 200, for example, one the user interfaces illustrated in Figs. 2 - 8 above, is redeployed as it was prior to deployment of the contextual user interface associated with the selected database object.

According to embodiments, once the user selects a different data object, the logical grouping of functionality controls may be dynamically changed again so that functionality controls under the selected mode of operation are once again changed in association with the different selected data object. For example, if the user selects a report object, the functionality controls deployed in the user interface 200 may be dynamically changed so that controls associated with report features of the associated database application 106 are dynamically displayed in the user interface 200 to provide those features commonly used when operating on a database report. Thus, the user experience is enhanced because the user is not required to search for particular functionality controls associated with the selected object. Instead, upon selection of a particular data object, those functionality controls associated with those features commonly used with respect to the selected object are automatically displayed to the user.

Fig. 9 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing contextual tabs for providing commands for formatting and layout for database forms. According to embodiments of the present invention, these two tabs provide the functionality commands needed for building form objects in a layout mode of an associated database application. A first type of data object for which contextual user

interfaces may be provided is a forms data object. Upon selection of a forms data object (i.e., a form), contextual user interface functionality may be provided for both layout and design of the selected form. Additionally, contextual user interface functionality for layout and design may also be provided for pivot chart and pivot table modes within the overall mode of forms. According to these embodiments, if a form object is selected, two additional task-based tabs 910, 915 for "Formatting" and "Control Layout" are deployed along the upper edge of the user interface 200, as illustrated in Fig. 9. If the user selects the "Formatting" tab 910, the logical groupings of functionality controls previously displayed in the user interface 200 are dynamically replaced with one or more logical groupings of functionality controls 920, 930, 940, 950 associated with applying formatting functionalities of the associated database application 106 to the selected form object.

The formatting contextual user interface provides functionality controls for quick formats, fonts, gridlines, fields and controls, and other groupings. For example, a first logical grouping of functionality controls 920 is deployed under the heading "Quick Format" for allowing a selection of one or more formatting types to the selected forms object. A second logical grouping of functionality controls 930 is deployed under the heading "Font" for providing functionality controls for applying various font properties to the selected form object. As illustrated beneath the user interface 200, one or more menus 960, 970 may be provided for displaying additional functionality controls for applying additional formatting properties to the selected form object. For example, selection of one of the gridlines controls illustrated in the menu 970 will cause an automatic application of the selected gridlines style to the selected form object. As should be appreciated, once a different data object is selected, for example, a report or table object, the functionality controls provided in the context of the previously selected form object may be replaced with functionality controls associated with the context of a subsequently selected data object.

Referring to Fig. 10, a different version of the contextual user interface 200 illustrated in Fig. 9 is provided for a selected report object. As illustrated in Fig. 10, contextual tabs are shown for providing commands for formatting and layout for database reports. According to an embodiment, these two tabs provide the functionality commands needed for building report objects in the layout mode of an associated database application. The user interface 200 for providing layout formatting to a selected report object provides an additional logical grouping 1030 for applying "Grouping and Totals" properties to a selected report object.

Fig. 11 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing contextual tabs for providing commands for database form and database

report layout. The form and report layout tab provide functionality commands used to layout a form or report in a database application layout mode. Upon selection of the "Control Layout" tab 915, the logical groupings of functionality controls previously displayed in the user interface 200 are dynamically replaced with one or more logical groupings of controls associated with control layout properties of the selected form object. That is, the functionality controls dynamically displayed in the user interface 200 provide access to functionality of the database application 106 for allowing the user to manipulate the location, the groupings, and the orders of controls on the selected database form. As illustrated in Fig. 11, the logical groupings of functionality controls are provided under such groupings as "Tools," "Control Layout," "Alignment," "Position" and "Show/Hide" for manipulating the locations, ordering and grouping of functionality controls provided in the selected form object.

Fig. 12 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing a contextual tab for designing a database form in a database design mode. The user interface illustrated in Fig. 12, includes commands/controls required for adding controls to a form object. This contextual user interface may be deployed in response to the selection of a particular data object, such as a form object, includes an additional top-level, task-based tab 1210 for applying design properties and functionalities to the selected form. In response to a selection of the "Design" tab 1210, one or more logical groupings of functionality controls are provided, including the "Tools" grouping, the "Font" grouping 1230, the "Grouping" grouping 1240, the "Gridlines" grouping 1245 and the "Fields and Controls" grouping 1250. Under these groupings, one or more functionality controls are displayed and are associated with functions of the database application 106 for applying design characteristics and properties to the selected form object.

Fig. 13 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing a contextual tab for providing commands for building a database query in a database design view. Upon selection of a database query for editing or further development, the user interface 200 may be dynamically populated with a "Design" tab 1210 for selectively deploying one or more logical groupings of functionality controls for providing functions of the database application 106 for building a database query. Upon selection of the "Design" tab 1210, a first logical grouping 1310 provides one or more functionality controls for designing a database query. A second logical grouping 1320 is populated under the heading "Query Type" for selecting a type of database query for building or editing. Additional logical groupings 1330, 1340 provide functionality controls for

applying database queries to rows and columns and for controlling the display properties of the selected database query.

Fig. 14 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing a contextual tab for providing commands for building a macro in a database design view. If a macro is selected via the database application 106, the user interface 200 may be populated with one or more logical groupings of functionality controls associated with applying functions of the database application 106 for designing, building and/or editing a selected macro. As illustrated in Fig. 14, a first logical grouping of controls 1410 provides database application tools for application to a selected macro, and logical groupings 1420, 1430 provide functionality controls associated with the insertion or deletion of rows and the application of display properties associated with a selected macro.

Fig. 15 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing a contextual tab for building a relationship between tables or queries in a database design view. This contextual user interface is provided upon the selection of a relationship between objects of a given database file. According to an embodiment, the user interface tab 1210 may be rendered from selecting a command on another tab, for example, the Advanced Tools tab 240. This opens a Relationship tab to provide commands for building relationships between objects and is not opened because of the context of an object. Upon selection of the "Design" tab 1210, logical groupings 1510, 1520 are provided for applying relationship tools, including editing relationships, creating relationship reports and applying display characteristics to relationships between database objects.

Fig. 16 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing a contextual tab for providing commands for building a pivot table in a database design view. Upon selection of a pivot table database object, the user interface 200 may be populated with the "Design" tab 1210, as illustrated in Fig. 16, and upon selection of the "Design" tab 1210, the user interface 200 is populated with one or more logical groupings of functionality controls for obtaining functionalities of the associated database application 106 for operating on a pivot table object. That is, upon selection of a pivot table object via the database application 106, the user interface 200 is automatically and dynamically populated with functionality controls for providing a user quick access to the functions of the associated database application 106 applicable to the selected pivot table object. For example, a "Tools" grouping 1610 is provided for displaying functionality controls associated with building and manipulating a pivot table object. Other logical groupings include the "Show/Hide" grouping 1620, the "Group" grouping 1630, the "Data" grouping 1640, the

“Active Field” grouping 1650 and the “Field and Sort” grouping 1660. In addition, a number of menus 1665 are displayed beneath the user interface 200 for providing additional functionality controls and options available to the user for operating on the selected pivot table. As should be appreciated, the menus 1665 may be deployed in response to the selection of a given control in the user interface 200.

Fig. 17 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing a contextual tab for providing commands for building a pivot chart in a database design view. The user interface 200 is illustrated containing contextual user interface functionality associated with a selected pivot chart object. Upon selection of a pivot chart object, the “Design” tab 1210 is deployed, and upon selection of the “Design” tab 1210, the logical groupings of functionality controls for applying pivot chart functions of the associated database application 106 are automatically deployed in the user interface 200 so that the user does not have to open one or more menus to search for functionality controls associated with pivot charts. As illustrated in Fig. 17, a “Tools” logical grouping 1710 is provided, a “Show/Hide” grouping 1720 is provided, a “Data” grouping 1730 is provided, an “Active Field” grouping 1740 is provided, a “Type” grouping 1750 is provided and a “Filter and Sort” grouping 1760 is provided in which a variety of functionality controls for operating on a selected pivot chart are made available to the user.

Fig. 18 illustrates a computer screen display showing the user interface illustrated in Fig. 3 and showing a contextual tab for providing commands for building a Table in a database design view. The user interface 200 is populated with contextual user interface functionality associated with a selected database table object. That is, upon selection of a table object via the database application 106, the “Design” tab 1210 is deployed and upon selection of the “Design” tab 1210, the user interface 200 is automatically populated with one or more logical groupings of functionality controls associated with applying table object functionality to a selected table object according to the functions provided by the associated database application 106. Logical groupings of controls illustrated in Fig. 18 include a “Tools” grouping 1810, a “Rows” grouping 1820 and a “Sharepoint List Settings” grouping 1830. As with the other contextual user interfaces described herein with reference to Figs. 9 - 18, selection of a different database object from the presently selected database object may cause the deployment of contextual user interface functionality associated with the subsequently selected data object.

As should be appreciated, according to one embodiment, selection of a given database object, as described herein with reference to Figs. 9 - 18, automatically causes deployment of

the logical groupings of functionality controls in the user interface 200 without the selection of the associated top-level, task-based tab if only one contextual user interface is available for the selected object. On the other hand, if two or more potential contextual user interfaces are available for a selected object, for example, formatting, control layout, or design, then additional top-level, task-based tabs, for example, the "Formatting" tab 910 and the "Control Layout" tab 915 (illustrated above in Fig. 9) are deployed for allowing selective deployment of functionality controls of the associated database application 106 under a desired task-based mode (for example, formatting versus control layout).

As should be appreciated, the illustrations and descriptions of contextual user interfaces described above with reference to Figs. 9 - 18 are for purposes of illustration and example only and are not limiting of the different types of contextual user interfaces and different types of functionalities that may be provided in response to the selection of a given database object. For example, according to the ACCESS® database application manufactured and marketed by MICROSOFT CORPORATION, certain database file types known as Access Data Projects (ADPs) are database types that require different sets of commands than other database files. According to embodiments of the present invention, selection of an ADP database object may cause the deployment of a contextual user interface, such as those illustrated and described with reference to Figs. 9 - 18, wherein particular functionality controls are provided for applying functions of the database application to ADP database objects. Likewise, contextual user interfaces may be developed and deployed for providing particular functionality associated with other types of database objects.

The user interface 200 illustrated in Fig. 19 represents an alternative embodiment of the user interface illustrated in Fig. 3. The user interface 200 illustrated in Fig. 19 provides top-level functionality tabs 1910, 1920, 235, 1930 for causing a display of logical groupings of buttons and controls for applying functionality of the associated database application to database objects and/or files. As illustrated in Fig. 19, upon selection of the "Home" tab 1910, a number of logical groupings of functionality controls are provided similar to the controls illustrated in Fig. 3 above. For example, logical groupings of controls are illustrated for views, clipboard functionalities, font functionalities, text functionalities, records functionalities, sorting and filtering functionalities, window functionalities and find functionalities.

Fig. 20 is an illustration of a computer screen display showing the user interface illustrated in Fig. 19 and showing a selected Create tab for providing commands for creating and/or inserting a database object in a database file. As illustrated in Fig. 20, the "Create" tab

1920 is selected for providing logical groupings of functionality buttons or controls for creating and inserting data objects into a database, for example, tables, forms, reports, queries, macros and database modules. The functionality buttons and controls provided in response to selection of the "Create" tab 1920 are similar to the functionality buttons or controls described above with reference to Fig. 4 provided in response to a selection of the "Insert" tab 225.

Fig. 21 illustrates a computer screen display showing the user interface illustrated in Fig. 19 and showing a selected External Data tab. As illustrated in Fig. 21, the "External Data" tab 235 is selected for providing logical groupings of buttons and controls associated with external data. The logical groupings of buttons or controls illustrated in Fig. 21 are similar to those buttons and controls illustrated in Fig. 6 and provide functionality of the associated database application for retrieving external data from a number of sources, including saved data, data maintained in a shared database, data maintained or enabled from a different software application, and the like, for inserting or otherwise applying to a given database object or database file.

Fig. 22 illustrates a computer screen display showing the user interface illustrated in Fig. 19 and showing a selected Database Tools tab for providing commands for performing maintenance on a given database. As illustrated in Fig. 22, the "Database Tools" tab 1930 is selected for causing a display of logical groupings of buttons or controls similar to the buttons or controls illustrated in Fig. 7 in response to a selection of the "Advanced Tools" tab 240. The database tools provided in the logical groupings of buttons or controls illustrated in Fig. 22 allow for such advanced features as database analysis, database security functions, database macro utilization and database relationship creation and maintenance.

As described above with reference to Figs 9 - 18, a variety of contextual tabs may be provided in the user interface 200 for providing one or more functionality buttons or controls associated with a particular database object selected or focused-on in a database file. Fig. 23 illustrates a computer screen display showing the user interface illustrated in Fig. 19 and showing a contextual tab for providing database functionality available for application to a table object. As illustrated in Fig. 23, upon selection or focus on a database table, a "Table Tools" tab 2310 may be deployed in the user interface 200 for providing logical groupings of buttons or controls associated with functionalities of the database application that may be applied to a database table. For example, functionality buttons and controls are provided in the user interface 200 in response to a selection of the "Table Tools" tab 2310 for applying

fields and columns, data type and formatting and data object relationships to data contained in a database table.

Fig. 24 illustrates an alternative ribbon-shaped user interface to the user interface illustrated in Fig. 8B for displaying logical groupings of functionality buttons and controls associated with a "Print Preview" tab. As illustrated in Fig. 24, a number of logical groupings of controls are provided for choosing different print settings for an associated database object similar to those print settings illustrated above in Fig. 8B. For example, a "Page Layout" grouping of controls provides for selectively printing a database object in "Portrait" orientation or "Landscape" orientation and provides for setting margins and other page layout properties prior to printing a given database object or file.

According to embodiments of the invention, an extensibility model is provided for allowing a user to customize the user interfaces 200 according to their individual needs. According to one embodiment, customizations of user interfaces 200 are enabled by providing an application programming interface (API) method that allows software developers or users to submit customizations of a given user interface 200 to the database application for use with a given database or database object (e.g., document, form, report, etc.). In one embodiment, customizations submitted via the API method are in the form of Extensible Markup Language (XML) based user interface customizations that will cause a given user interface 200 to have custom properties when loaded. The customized XML markup need not reside in a given database or database object. Thus, developers may create database templates (e.g., forms or reports templates) having customized versions of the user interface 200 that may be distributed to users for implementation as software add-ins to the database application.

A developer of a user interface customization may create an AutoExec macro that calls a desired customization for loading. The AutoExec macro causes the desired customization to call the API method (e.g., an Application.LoadCustomUI method) and specifies the XML and any associated friendly name pairs for the customization. Upon receipt by the database application, a user interface cookie is generated by the database application for each unique customization name (e.g., sales form user interface), and customizations are submitted by the application to the user interface 200 for implementation. Each unique user interface customization may be stored in a library of customizations and may be presented to end users via a menu of choices (e.g., a combo box) from which a user may select a desired customization.

The loading and unloading of customizations is application driven via properties associated with the customizations for a given database or database object. Database level customizations take effect after database restart. For database objects, such as forms and reports, changes to a given user interface may be made while the database application is running by selecting a desired user interface customization for a given database object.

As described above, user interface customizations may be provided for various database objects, for example, forms and reports. In the case of contextual user interface tabs, as described above with reference to Figs. 9 - 18, XML markup is rendered such that it replaces the contents of the native user interface 200 when an associated object is selected or focused on. For example, when a report is opened for which a customized contextual user interface is provided, the contents of the native user interface 200 are replaced according to the XML markup associated with a customized user interface designed for the report.

For purposes of illustration, consider the following examples. A user generates and deploys a number of sample databases. The user's desire is to provide customized menus for her major forms and reports. For example, her sales and inventory application main navigation form provides a menu that allows a user to navigate to a Customers and Payments form; an Orders form; a Product and Inventory form; a Vendors and Purchase Orders form; and the like. According to an embodiment of the invention, a customized user interface 200 is generated, as described above, and when any of these forms are launched, a customized form-editing user interface 200 is provided.

According to another example, a large international organization may develop a number of database objects (forms, reports, etc.) in multiple languages (for example, German, English, Italian, French, etc.). According to embodiments of the present invention, a developer of such database objects may develop a customized user interface for each potential language for a given database object, for example, a database report. Upon launching the database report, a dialog may be presented to the user to ask the user which language is desired for the launched database report. Once the user selects a given language, the customized user interface associated with the selected language may be loaded, and the user interface 200 is launched with the desired language.

According to another example, security may be imposed on a given database or database object by customizing the commands available in the user interfaces 200 based on user security access. For example, a given user may be required to provide a password or other login credential to launch a given database or database object. A given user's login credentials may be associated with a version of one or more user interfaces 200 that provide a

customized set of commands for allowing the user to perform certain specified operations on the selected database or database object. Once the identified user's credentials are associated with an XML markup stored for the specified security level, the associated XML markup may be loaded, as described above, and the required user interface 200 will be launched with a set of commands prescribed by the associated XML markup.

These embodiments are also useful for purposes of backward compatibility because a particular user may have generated a large number of database objects, such as forms or reports, using a previous or legacy version of the database application where the previous version of the database application user interface provided functionality via command bars, toolbars, dropdown menus and the like. Prior or legacy user interfaces 200 may be stored as customized user interfaces and may be launched in place of current version user interfaces 200 upon command. According to one embodiment, the legacy user interface customizations are only launched in place of the current version user interfaces when the entire database application has been set to use backward compatibility user interfaces. Alternatively, the legacy user interface customizations may be rendered by selection of an "Add-Ins" type tab deployed in the user interfaces 200. According to another embodiment, a backward compatibility mode/view may be included in the current version user interfaces 200. If a given user had previously configured a database or database object not to allow full user interfaces 200, a different display of the a given user interface 200 may be provided where only a single tab is displayed for all database objects and contexts. Selection of the single tab may then cause the user interface 200 to be populated with legacy user interface commands as desired.

As described herein, an improved user interface is provided for exposing task-based top-level functionality tabs for displaying logical groupings of selectable database application functionality controls associated with given database application modes of operation. It will be apparent to those skilled in the art that various modifications or variations may be made in the present invention without departing from the scope or spirit of the invention. Other embodiments of the invention will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein.

Claims

WE CLAIM:

1. A computer readable medium containing computer executable instructions which when executed by a computer perform a method for providing functionality from a database application via an improved user interface, comprising:

providing a plurality of functionalities available from the database application;

organizing the plurality of functionalities according to one or more tasks that may be performed with the database application;

providing in the user interface a user interface tab for each of the one or more tasks;

and

upon receiving an indication of a selection of a first user interface tab, providing in the user interface one or more selectable controls for selecting one or more functionalities organized under a first task associated with the selected first user interface tab.

2. The computer readable medium of Claim 1, further comprising upon receiving an indication of a selection of one of the one or more selectable controls, applying functionality associated with the selected one of the one or more selectable controls to a selected document object.

3. The computer readable medium of Claim 1, further comprising upon receiving an indication of a selection of a second user interface tab, providing in the user interface one or more selectable controls for selecting one or more functionalities organized under a second task associated with the selected second user interface tab.

4. The computer readable medium of Claim 1, further comprising grouping the one or more selectable controls into one or more logical groupings of selectable controls where each of the one or more logical groupings is associated with a subset of functionalities of the database application associated with the selected user interface tab.

5. The computer readable medium of Claim 1, further comprising:
upon receiving an indication of a focus on a second user interface tab, providing in the user interface one or more selectable controls for selecting one or more

functionalities organized under a second task associated with the second user interface tab;
and

upon cessation of the focus on the second user interface tab, providing in the user interface the one or more selectable controls for selecting one or more functionalities organized under the first task associated with the first user interface tab.

6. The computer readable medium of Claim 5, further comprising:

after providing in the user interface one or more selectable controls for selecting one or more functionalities organized under a second task, receiving an indication of a selection of one of the one or more selectable controls for selecting one or more functionalities organized under a second task;

applying functionality associated with the selected control to a selected object;
and

if the second user interface tab is not selected for persisting a provision in the user interface of one or more selectable controls for selecting one or more functionalities organized under a second task, providing in the user interface the one or more selectable controls for selecting one or more functionalities organized under the first task associated with the selected first user interface tab.

7. The computer readable medium of Claim 1, further comprising:

receiving an indication of a selection of a data object for editing via the database application; and

upon receiving the indication of the selection of the data object for editing, replacing the one or more selectable controls in the user interface for selecting one or more functionalities organized under a first task with one or more selectable controls representing a first subset of a plurality of functionalities provided by the database application that may be used for editing the selected object.

8. The computer readable medium of Claim 7, prior to replacing the one or more selectable controls in the user interface for selecting one or more functionalities organized under a first task with one or more selectable controls representing a first subset of a plurality of functionalities provided by the database application that may be used for editing the selected object, further comprising providing in the user interface a contextual user interface

tab for each of one or more tasks that may be performed with the subset of the plurality of functionalities.

9. The computer readable medium of Claim 8, further comprising upon receiving an indication of the selection of the contextual user interface tab, providing in the user interface the one or more selectable controls in the user interface for selecting one or more functionalities organized under a first task with one or more selectable controls representing a first subset of a plurality of functionalities provided by the database application that may be used for editing the selected object.

10. The computer readable medium of Claim 9, further comprising grouping the one or more selectable controls representing a first subset of a plurality of functionalities provided by the database application that may be used for editing the selected object into one or more logical groupings of selectable controls whereby each of the one or more logical groupings is associated with a subset of the one or more selectable controls representing a first subset of a plurality of functionalities provided by the database application that may be used for editing the selected object.

11. The computer readable medium of Claim 7, wherein receiving an indication of a selection of a data object for editing via the database application, includes receiving an indication of a selection of a database form for editing via the database application.

12. The computer readable medium of Claim 7, wherein receiving an indication of a selection of a data object for editing via the database application, includes receiving an indication of a selection of a database report for editing via the database application.

13. The computer readable medium of Claim 7, wherein receiving an indication of a selection of a data object for editing via the database application, includes receiving an indication of a selection of a database table for editing via the database application.

14. The computer readable medium of Claim 7, wherein receiving an indication of a selection of a data object for editing via the database application, includes receiving an indication of a selection of a database pivot chart for editing via the database application.

15. The computer readable medium of Claim 7, wherein receiving an indication of a selection of a data object for editing via the database application, includes receiving an indication of a selection of a database pivot table for editing via the database application.

16. An improved user interface for providing functionality from a database application, comprising:

one or more tabs disposed in an upper row of the user interface, each of said one or more tabs identifying a mode of operation of the database application that may be utilized with the database application, and each of said one or more tabs being operative upon selection to cause a display in the user interface of one or more selectable functionality controls associated with an associated mode;

said one or more selectable functionality controls being disposed in a row underneath the disposition of the one or more tabs, each of said one or more functionality controls being operative to cause an application of an associated functionality of the software application to a selected object; and

said one or more selectable functionality controls being organized and displayed in the user interface in one or more logical groupings, whereby each of said one or more logical groupings is associated with a subset of functionalities available from the database application that may be applied to a selected object according to a portion of a mode of operation of the database application associated with a selected tab.

17. A method of customizing deployment of a database application user interface, comprising:

storing a customized database user interface for a database;

receiving an API method call that specifies the requested customized database user interface;

in response to receiving an API method call that specifies the requested customized database user interface, retrieving the customized database user interface; and

replacing a native database user interface with the customized user interface.

18. The method of Claim 17, wherein storing a customized database user interface for a database includes storing a customized database user interface for a specified database object.

19. The method of Claim 17, wherein storing a customized database user interface for a database includes storing Extensible Markup Language markup representing the customized database user interface.

20. The method of Claim 17, further comprising:
receiving a unique name for the customized database user interface;
wherein receiving the API method call that specifies the requested customized database user interface includes receiving an API method call that specifies the unique name for the customized database user interface;
creating a user interface cookie for the unique name; and
providing the unique name in a selectable menu of customized database user interfaces for allowing selection of the unique name for initiating the API method call that specifies the requested customized database user interface.

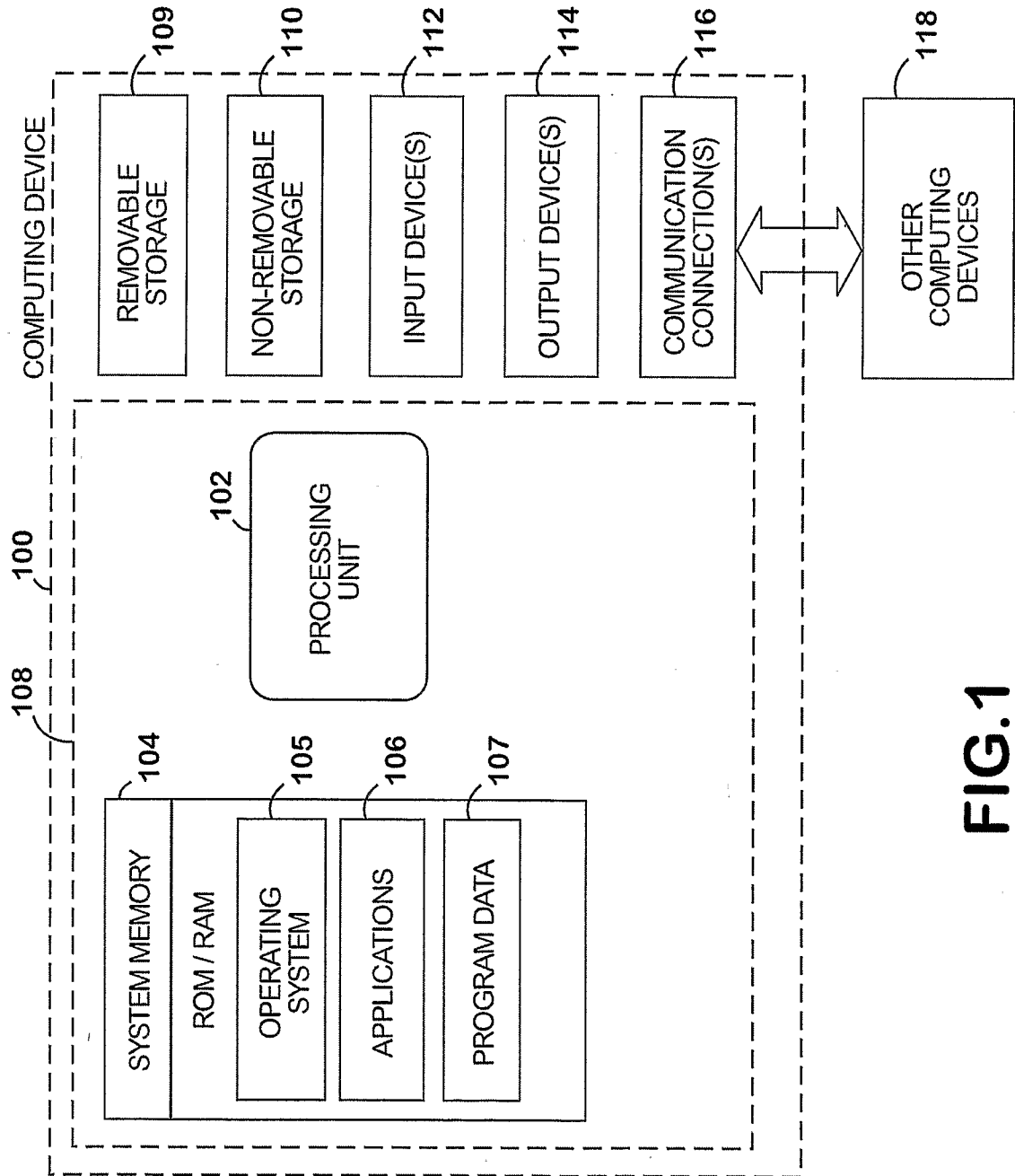


FIG.1

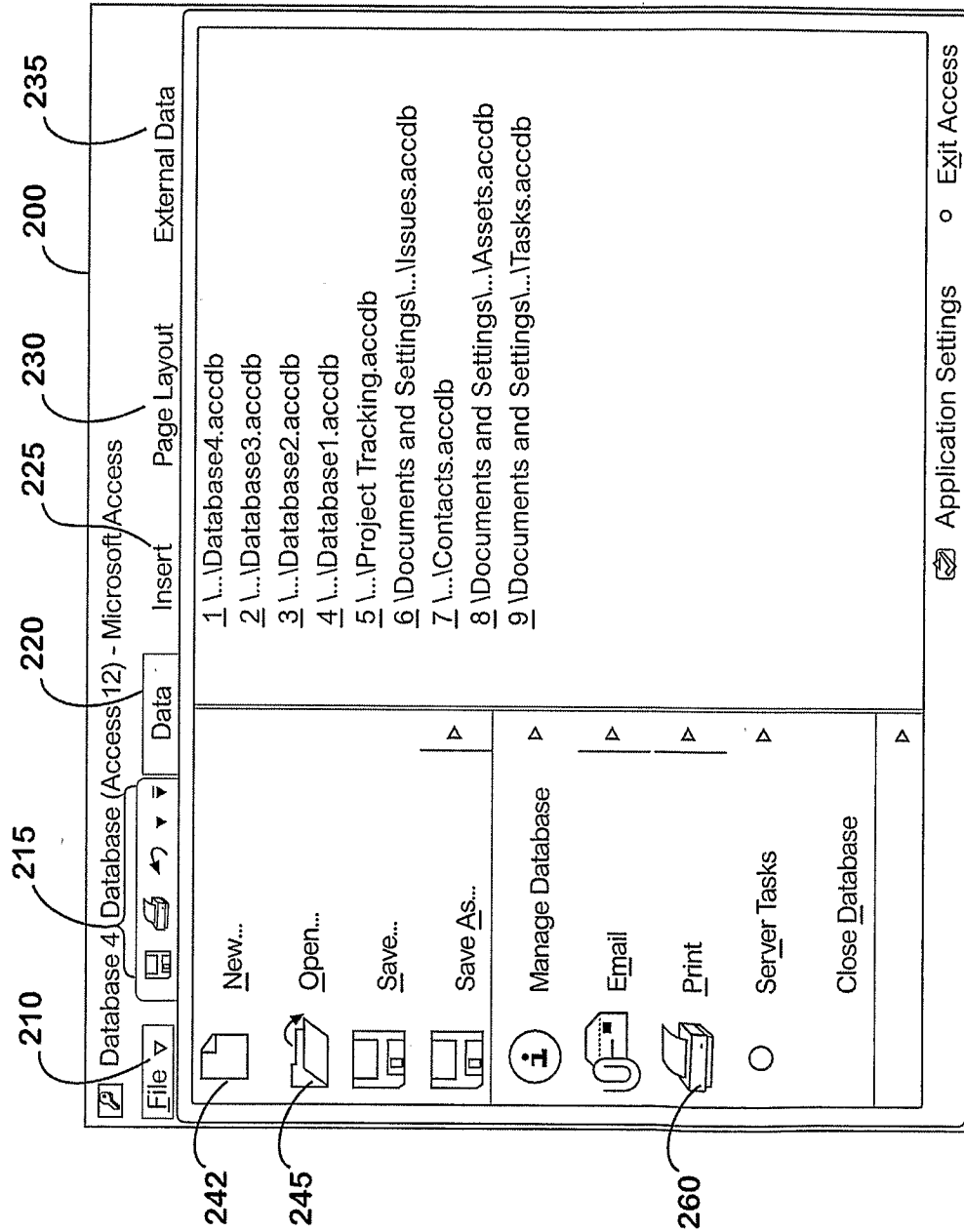


FIG.2

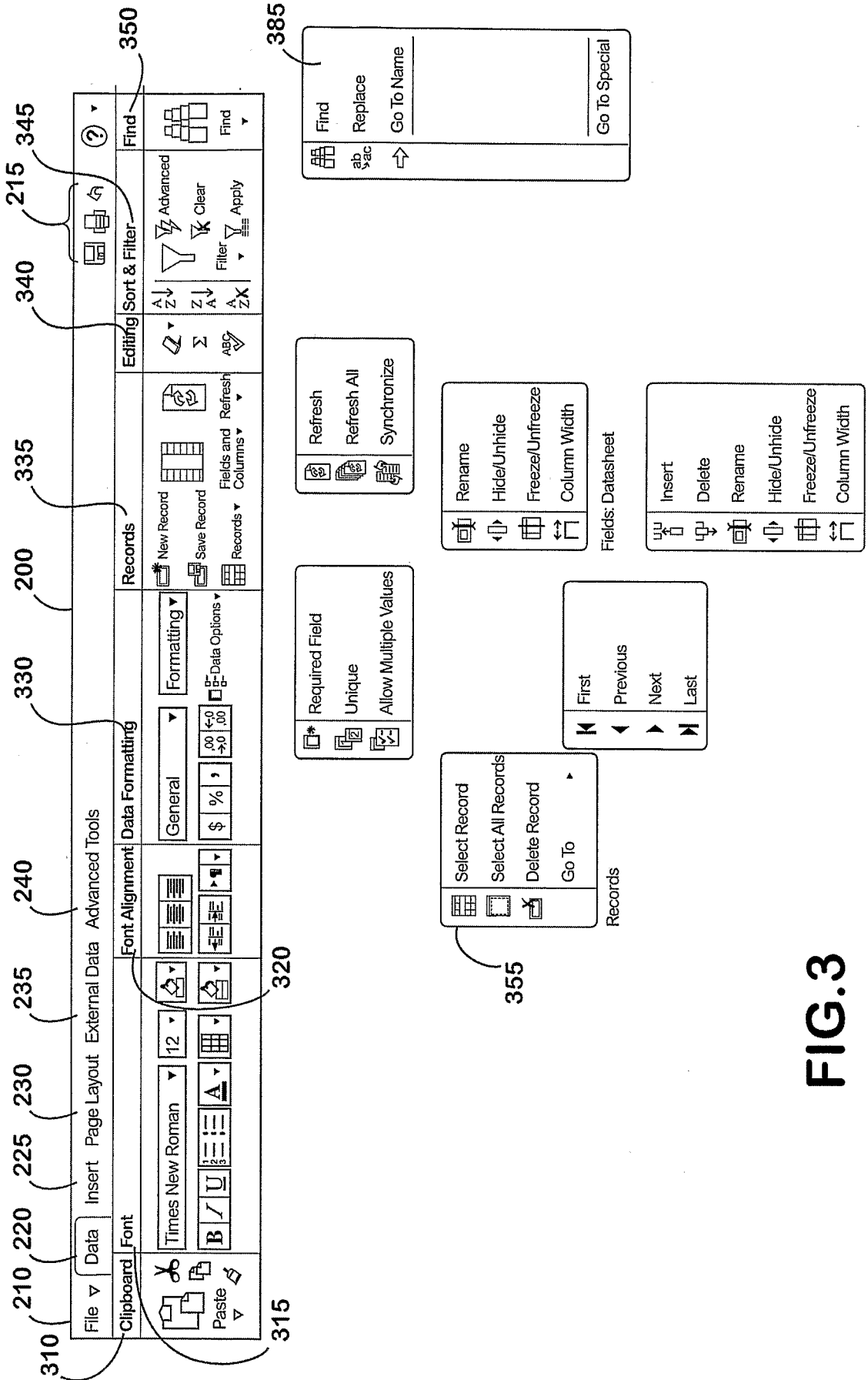


FIG.3

Fields: Table

Fields: Datasheet

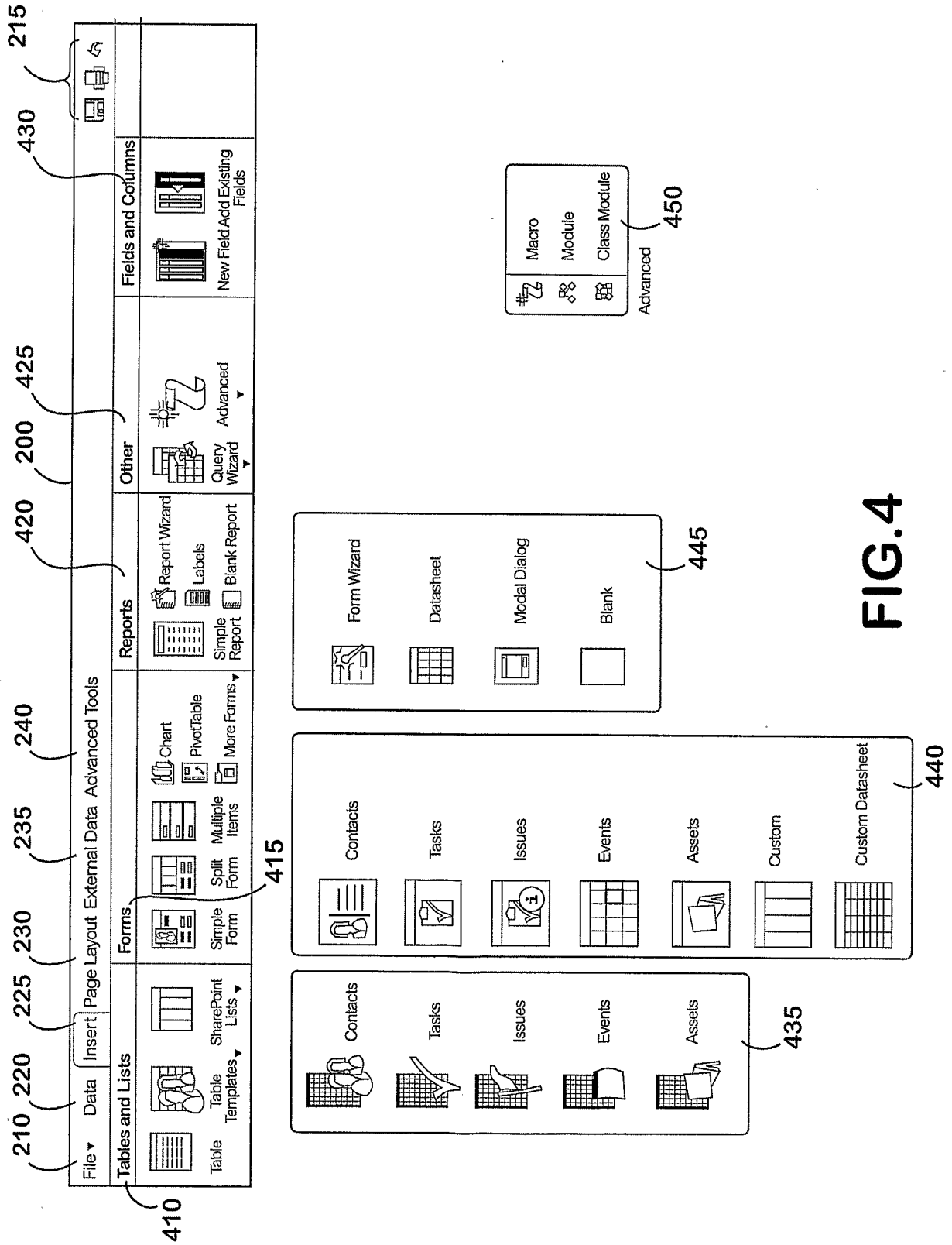


FIG. 4

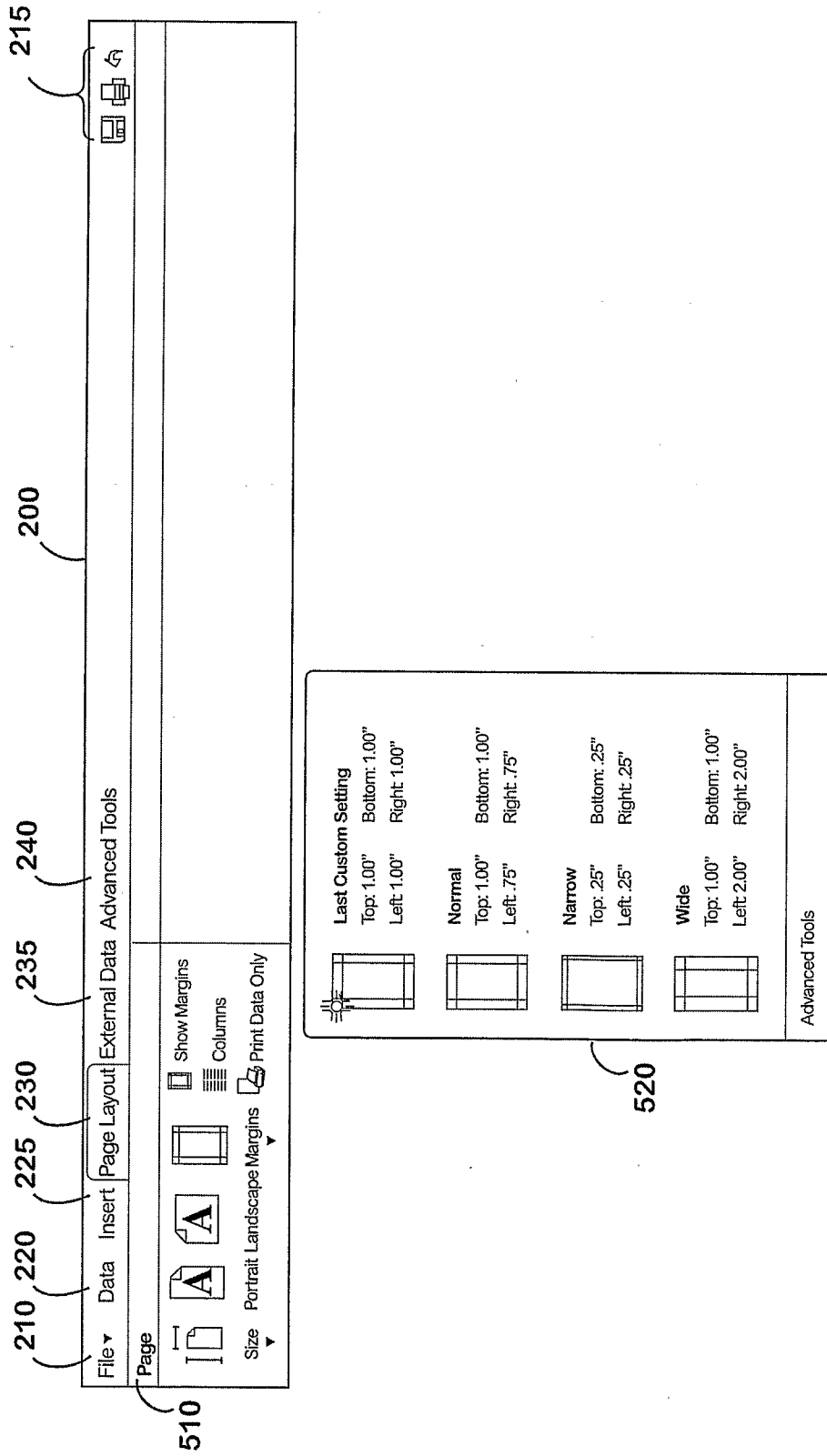


FIG.5

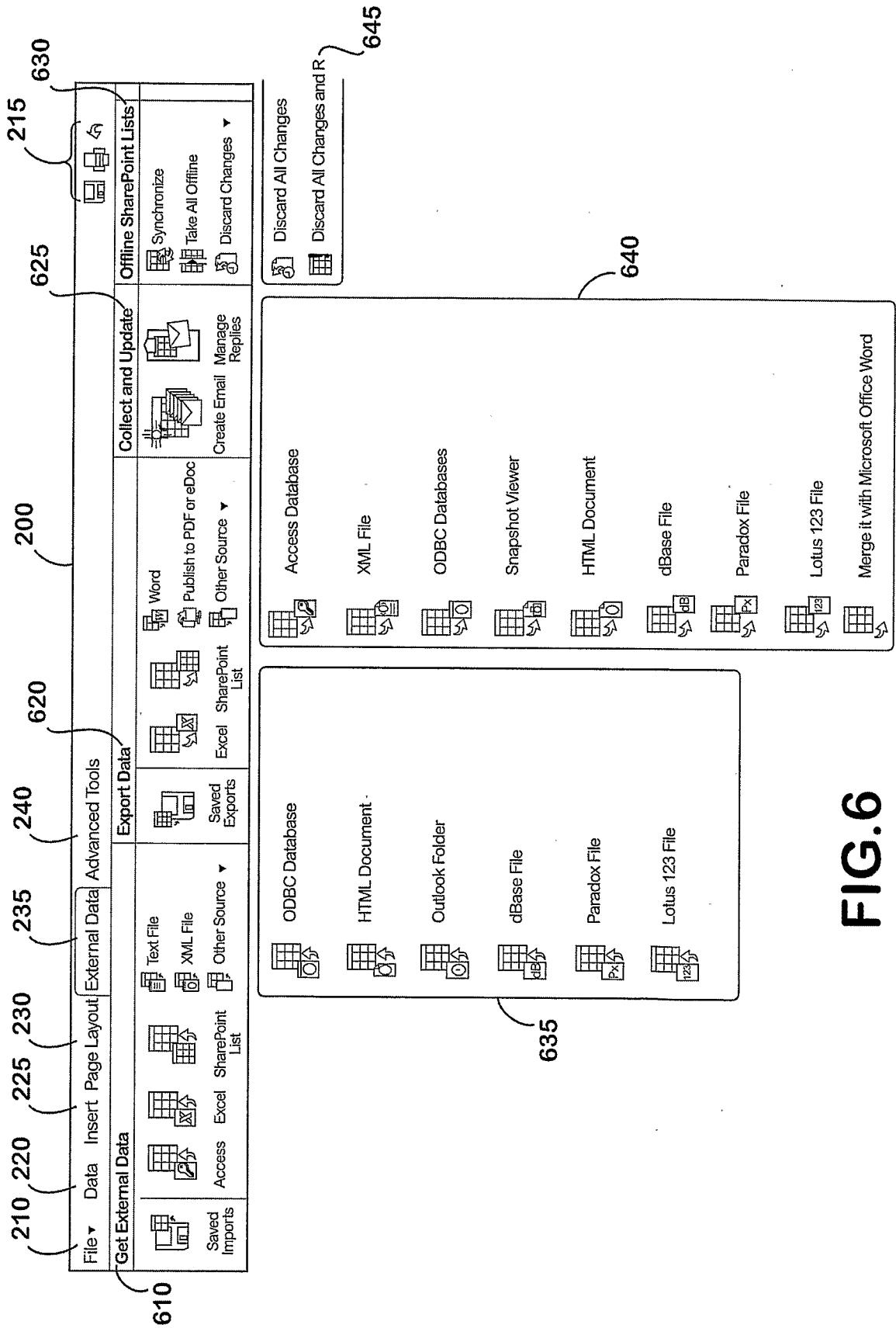


FIG. 6

7/25

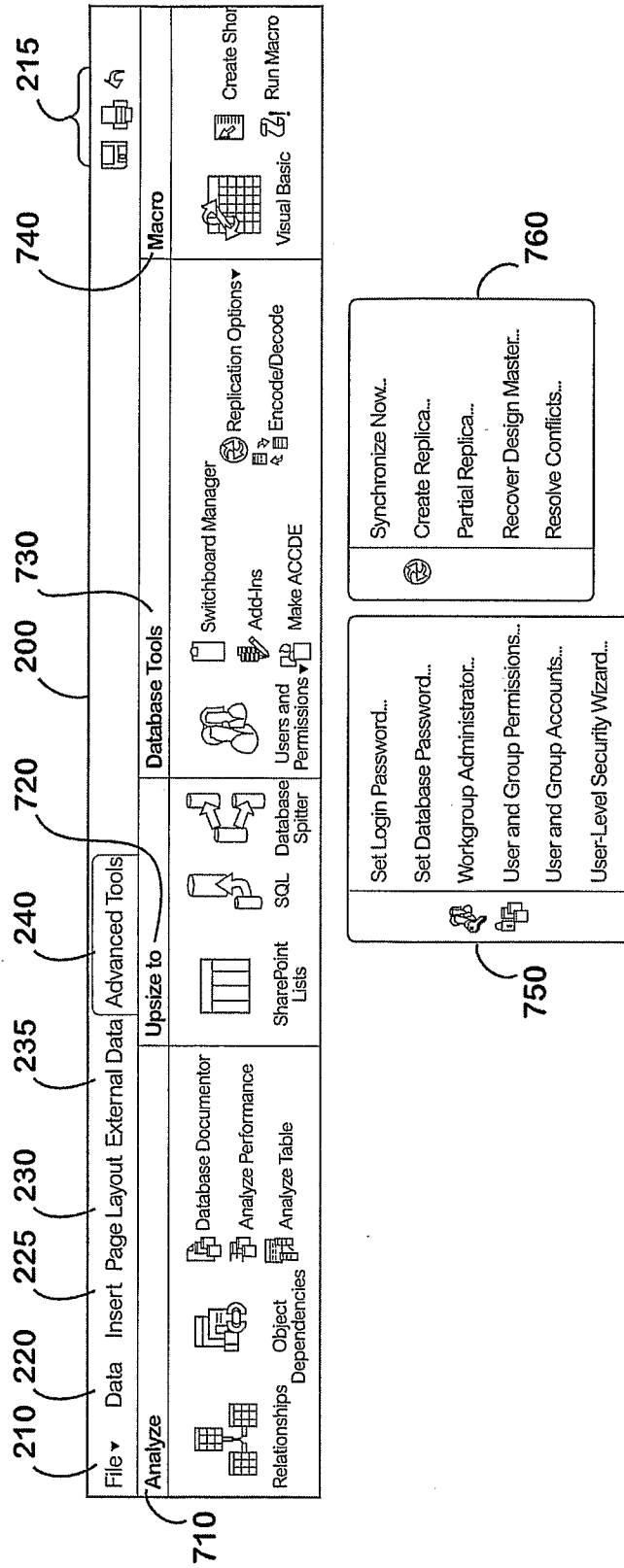


FIG. 7

8/25

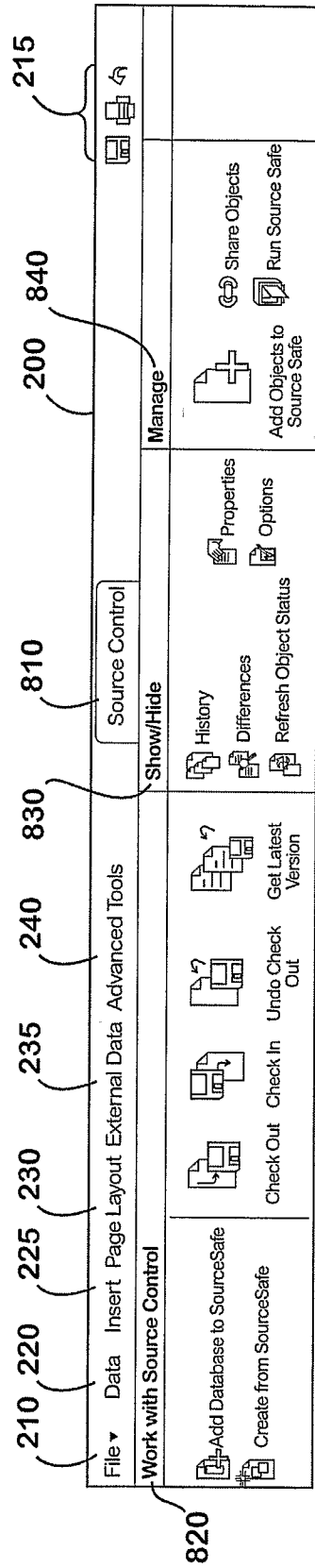


FIG.8A

9/25

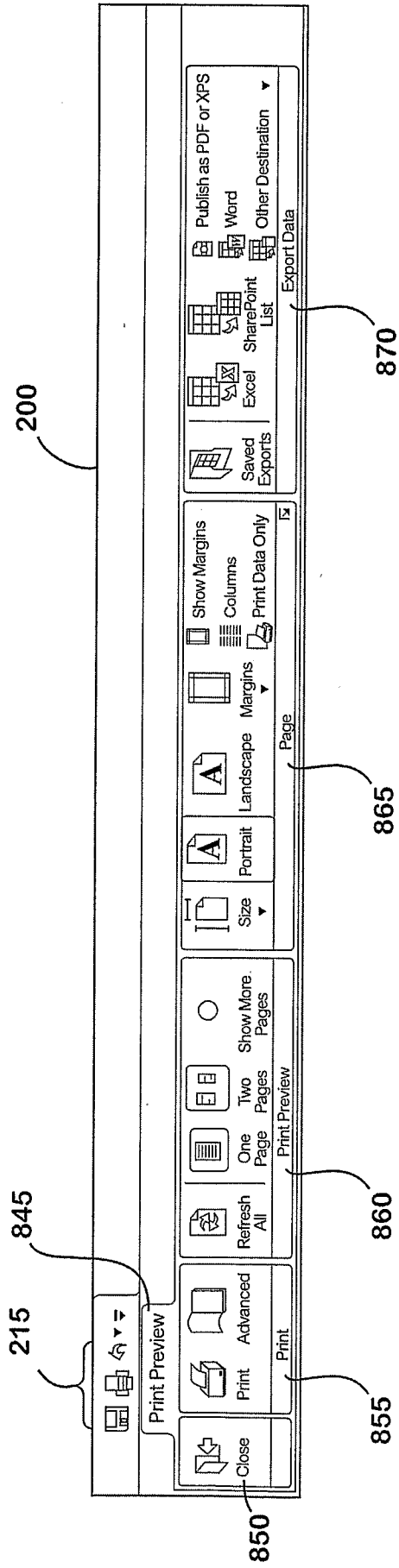
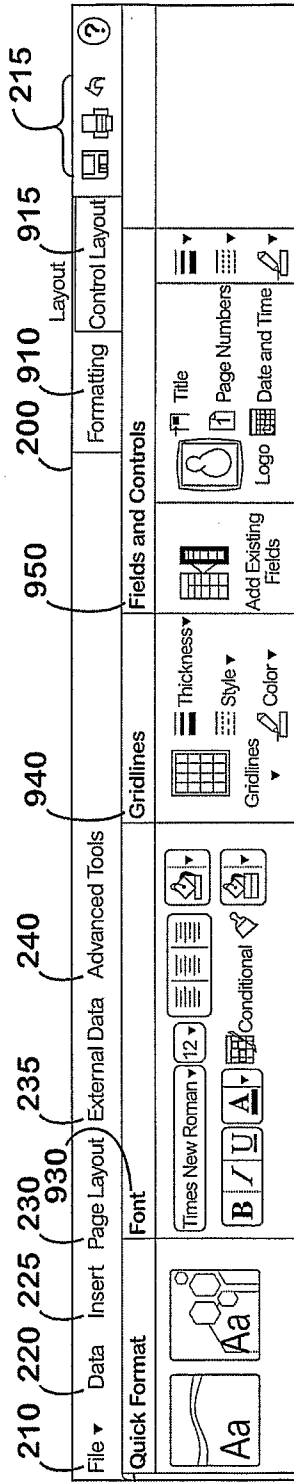
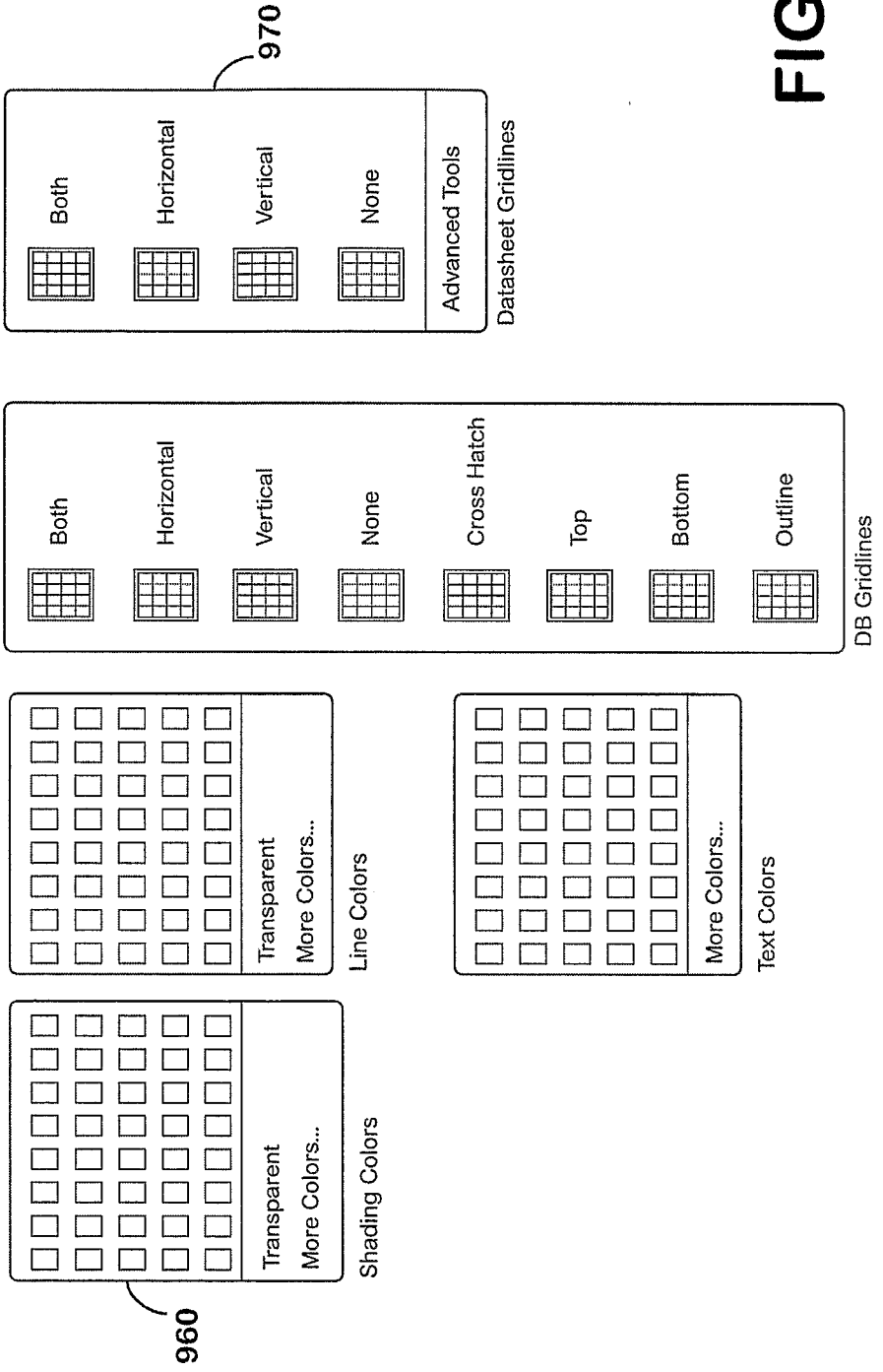


FIG.8B

10/25



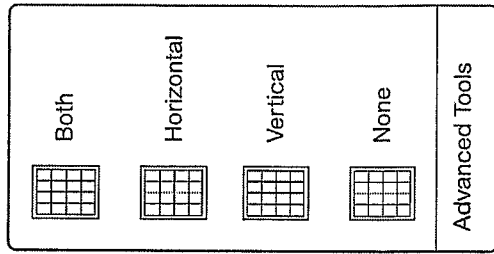
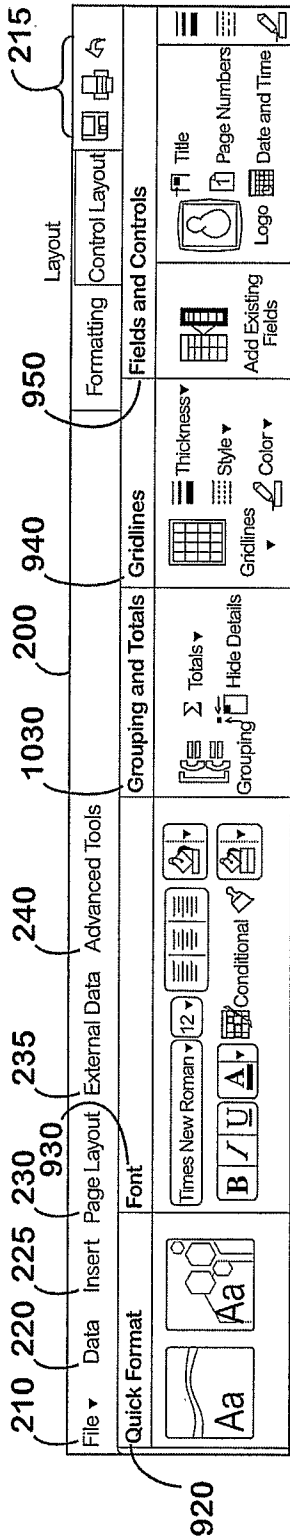
920



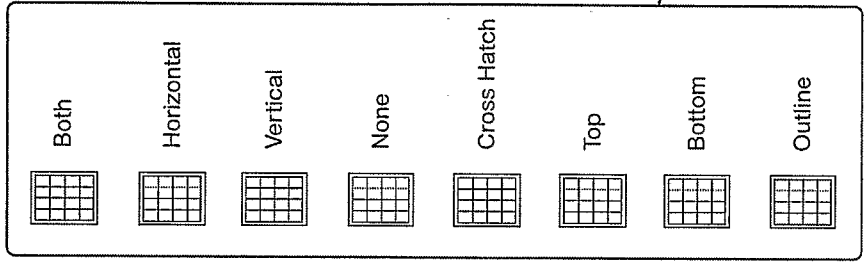
960

970

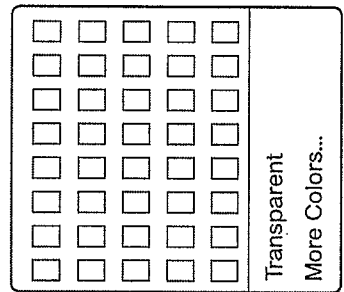
FIG. 9



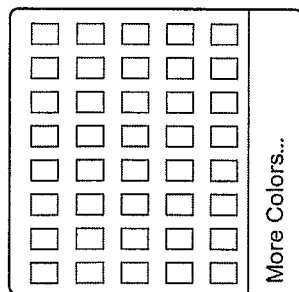
Datasheet Gridlines



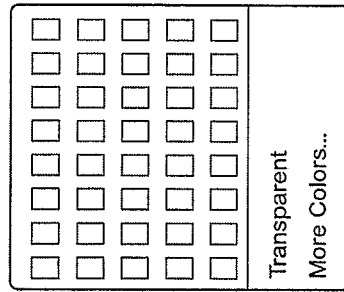
DB Gridlines



Shading Colors



Line Colors



Text Colors

1055

FIG.10

12/25

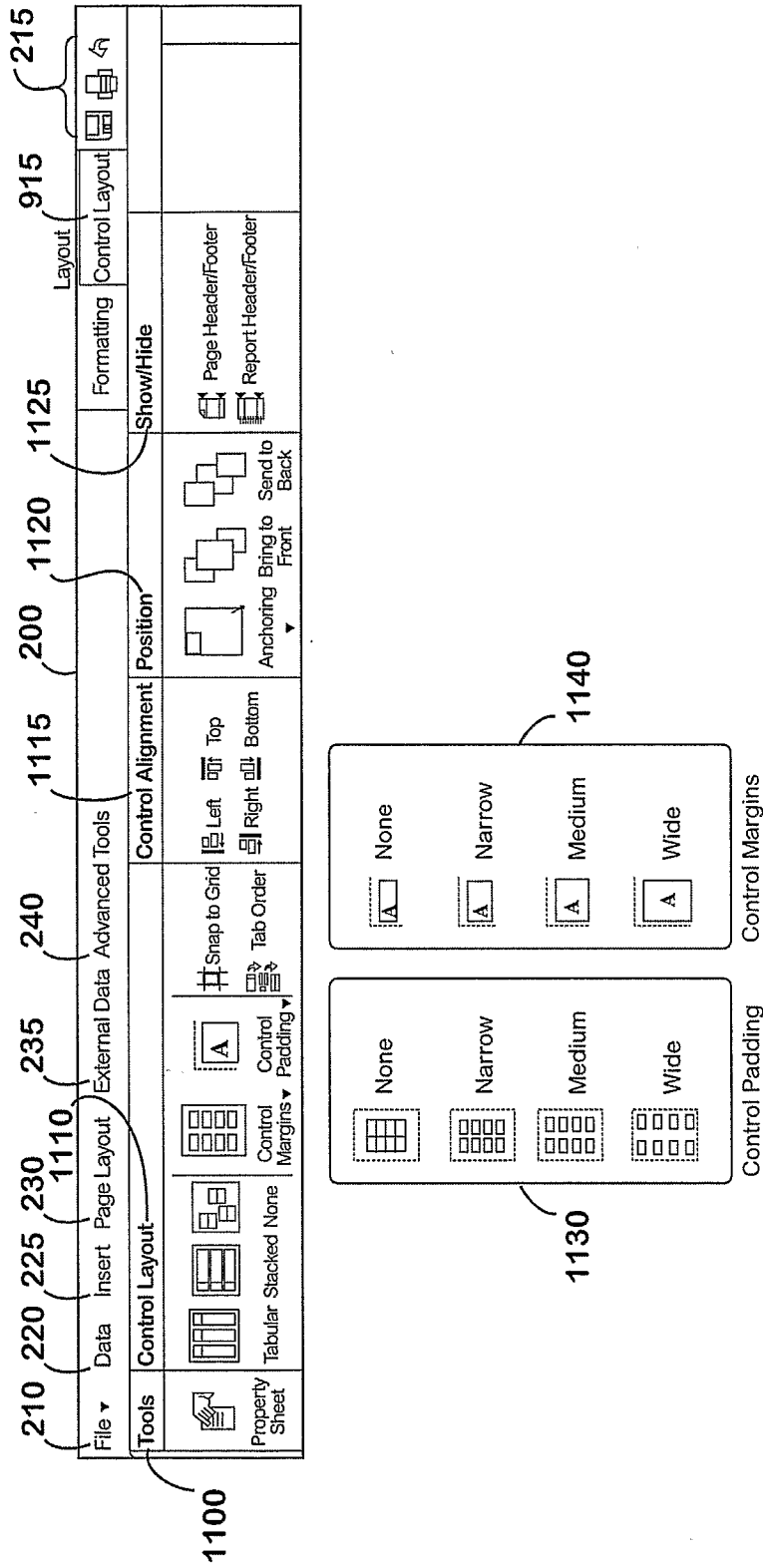


FIG.11

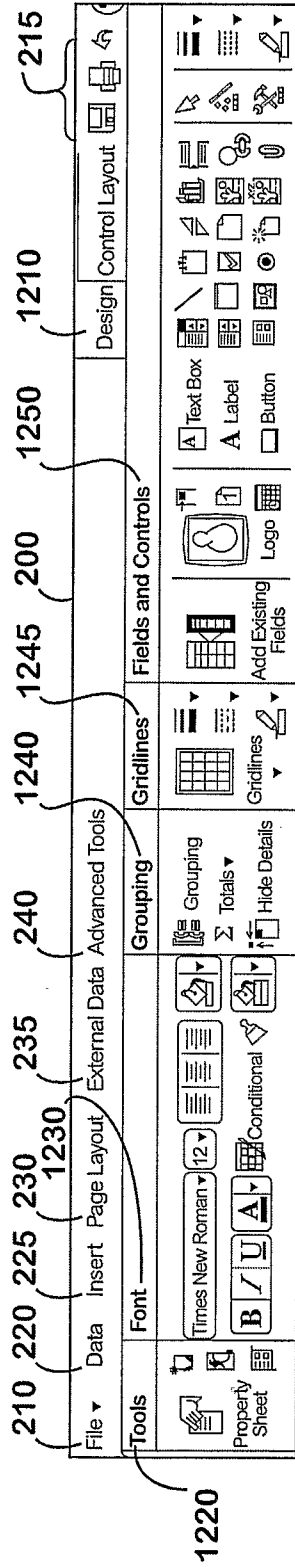


FIG.12

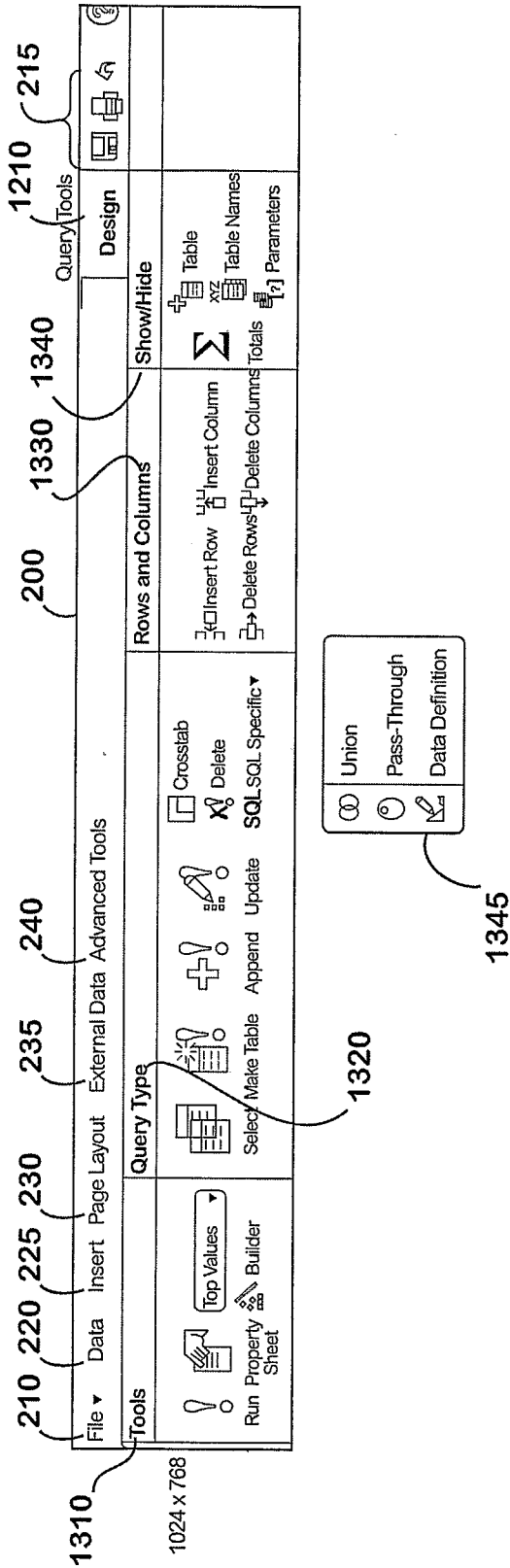


FIG.13

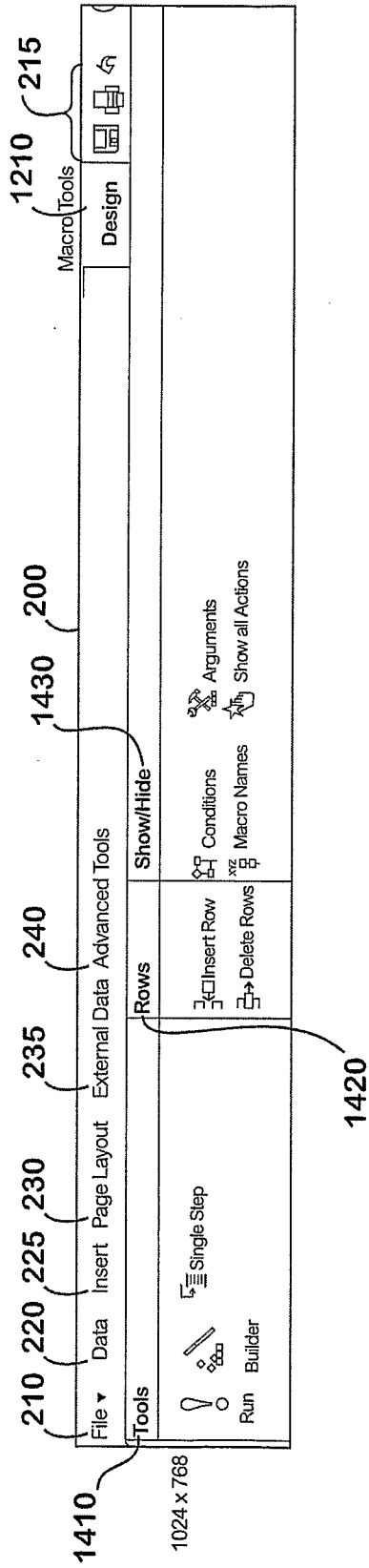


FIG.14

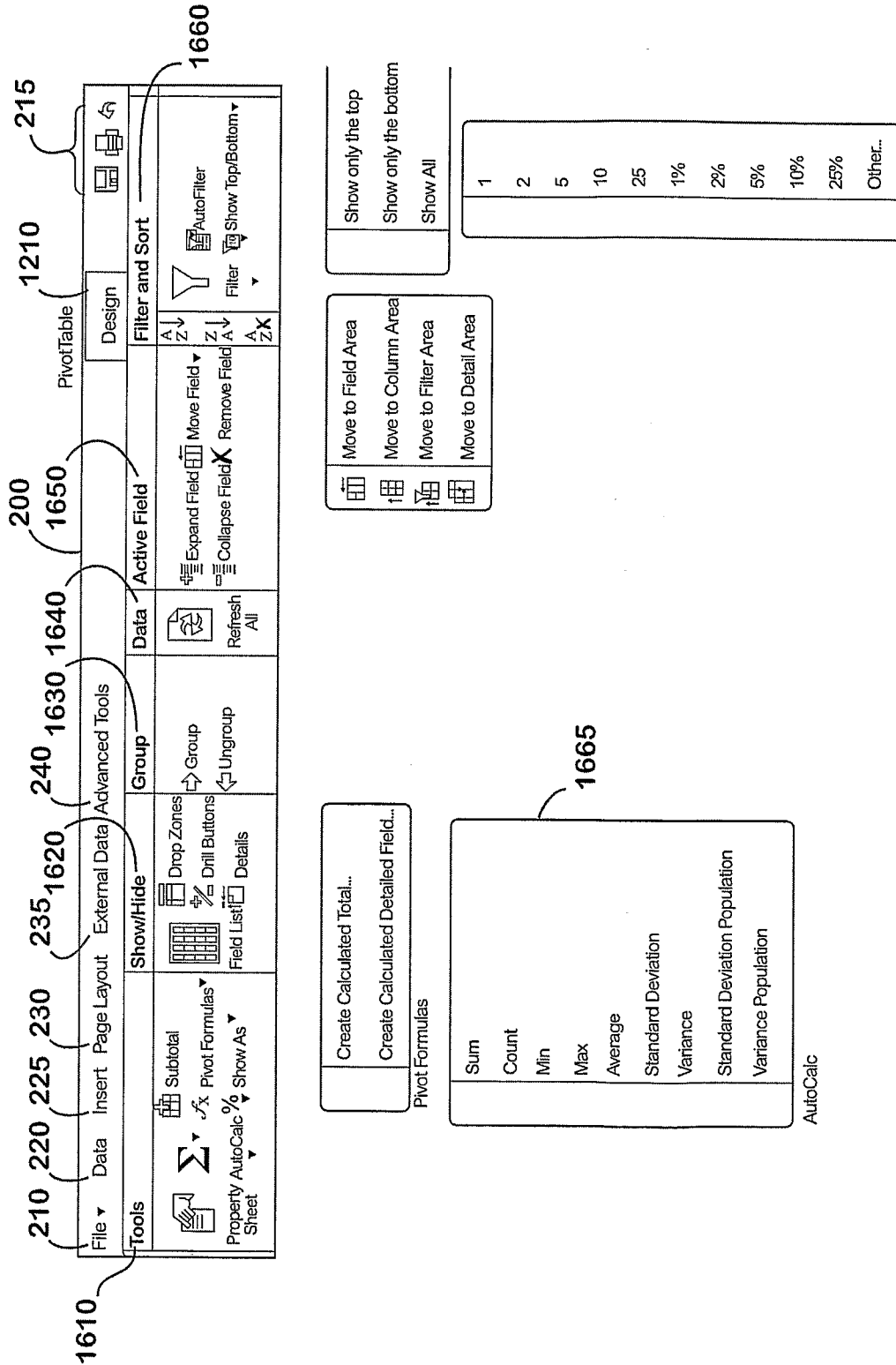


FIG.16

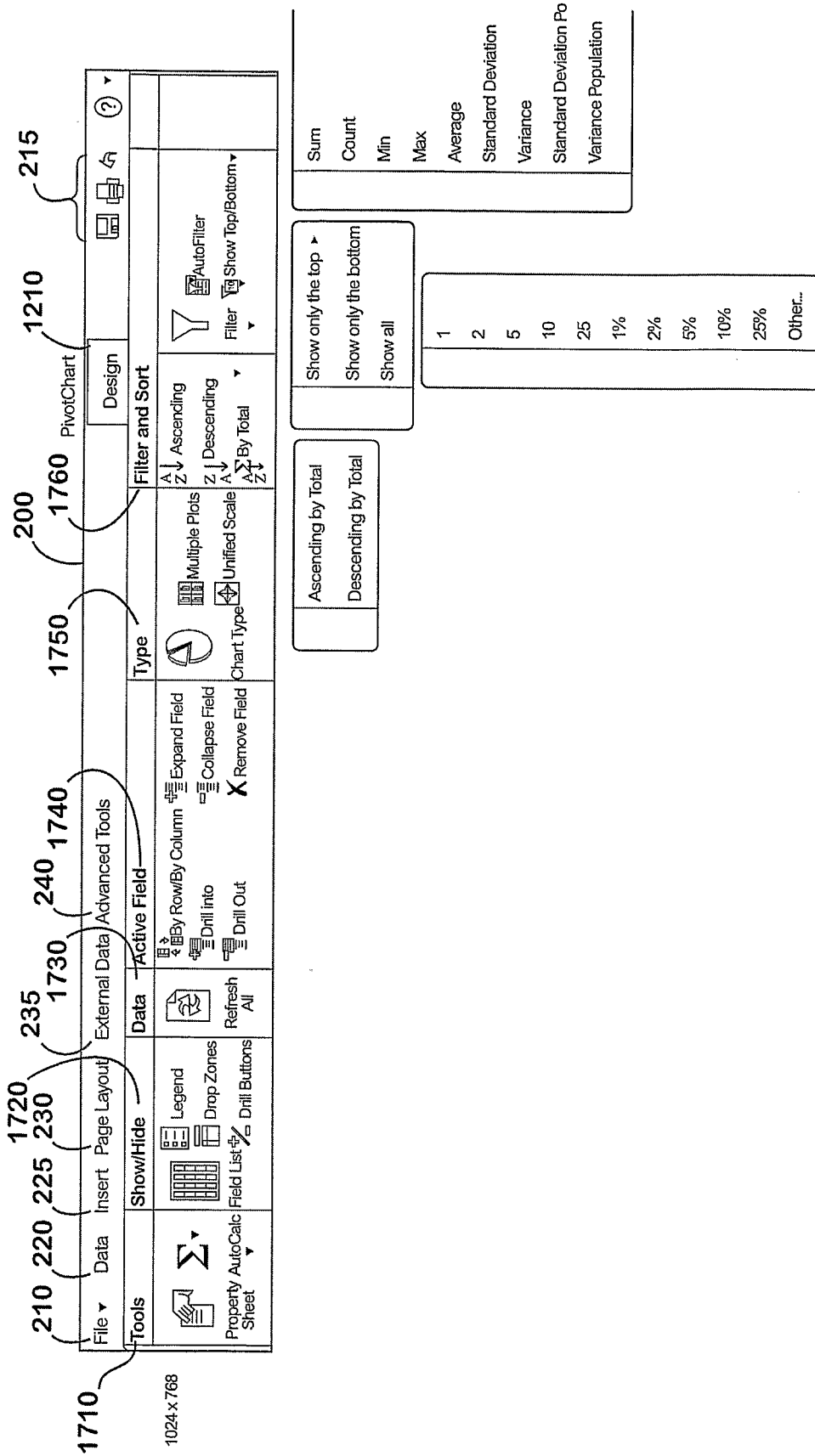


FIG.17

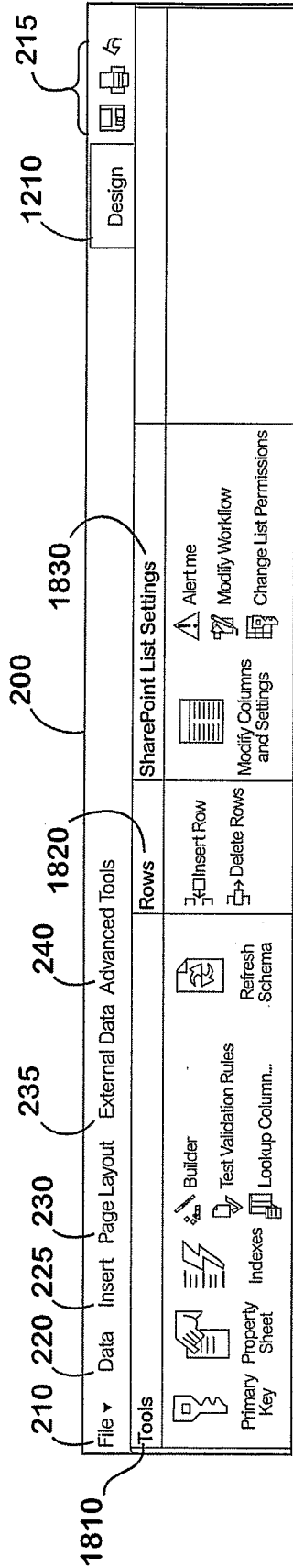


FIG.18

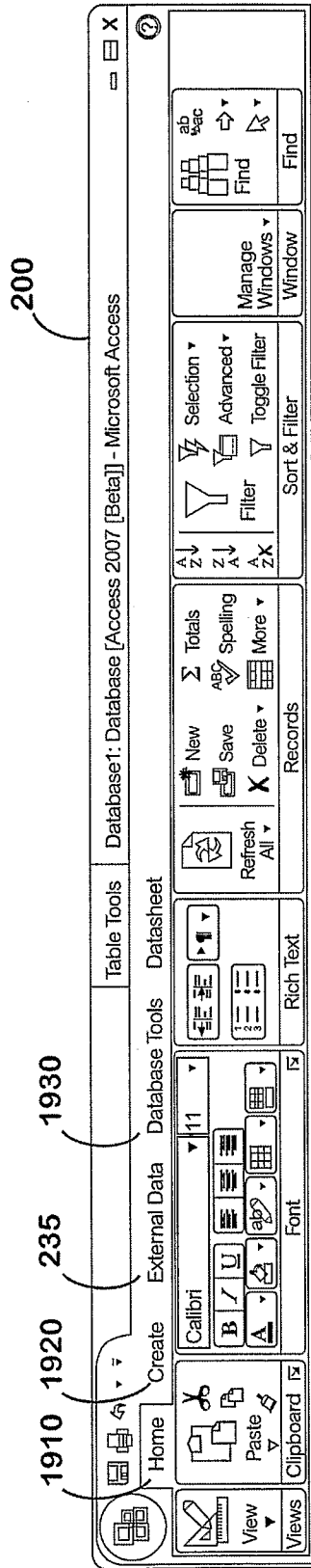


FIG.19

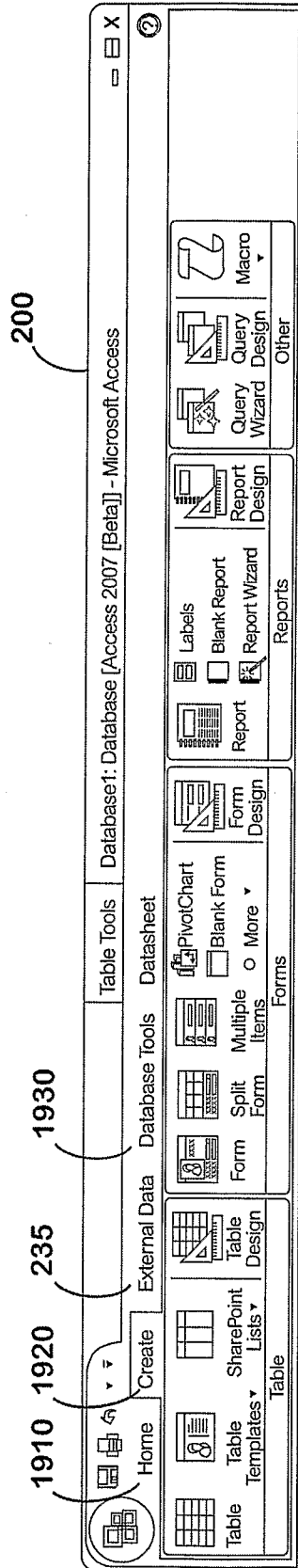


FIG. 20

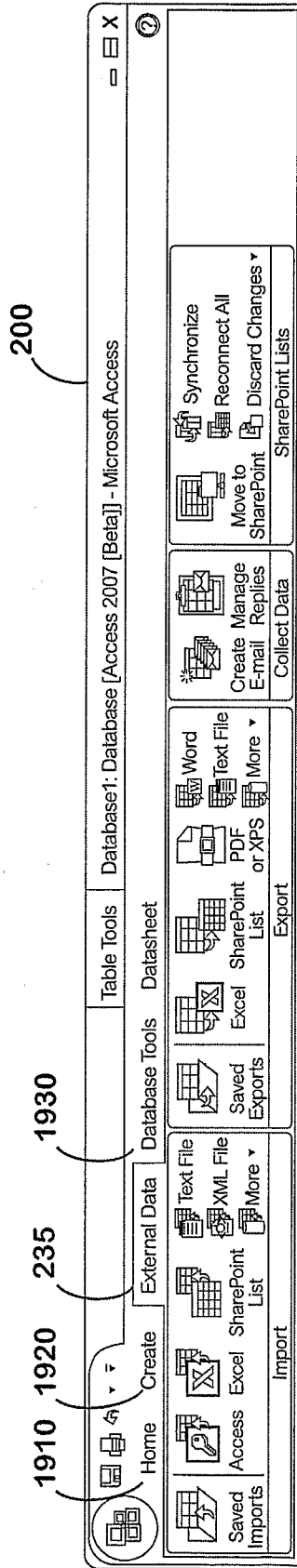


FIG. 21

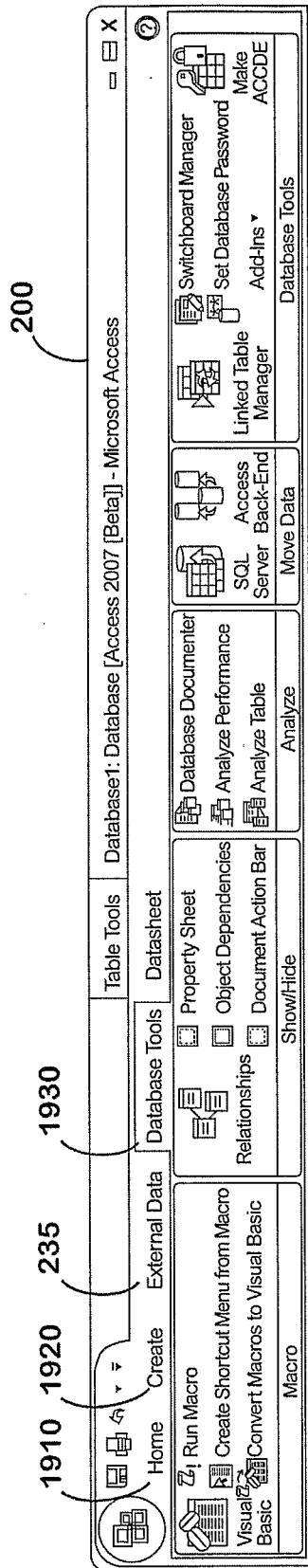


FIG.22

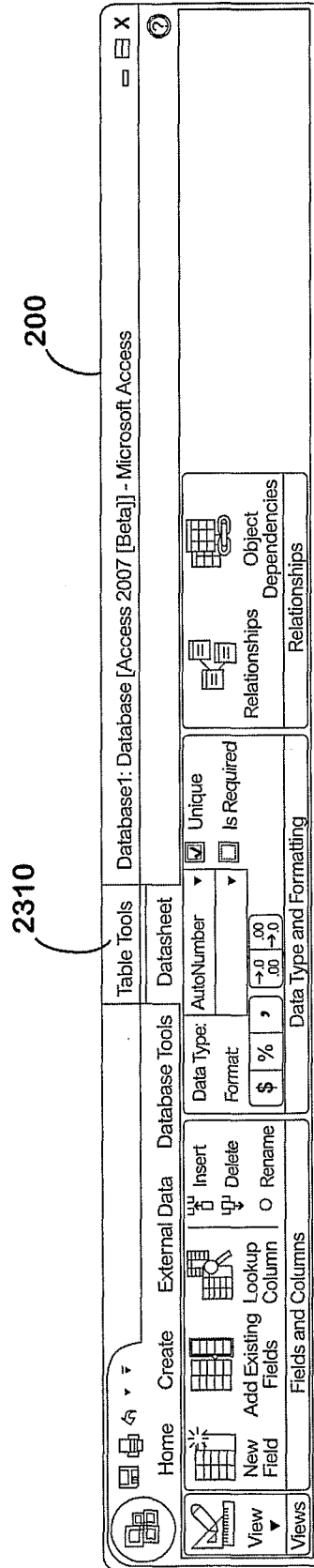


FIG.23

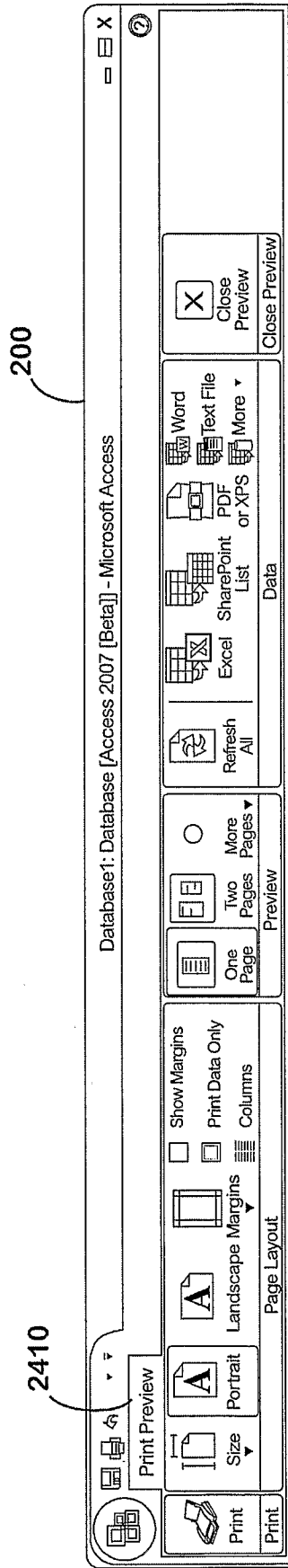


FIG.24