QlikView®

Introduction

CONTENTS

Background

Whether you work for a startup company or a 100-year old corporation, ***data*** is at the heart of your business. You have many different kinds of data: financial data, product data, customer data, supplier data, competitive data, and human resource data. This data is stored in many different ways: databases, spreadsheets, word processing documents, web pages and multimedia files.

Different software products and systems are used to collect and utilize the various types of data that keep your business running. Operational systems focus on gathering transactional data encountered each business day. Employees create spreadsheets, memos and reports that they wish to share with others to communicate business data. Online analytical processing (OLAP), data mining, ad hoc query and reporting, data warehousing, and corporate portals represent different ways of storing, managing, and working with business data.

Business runs on data and *your* business runs on *your* data. The purpose of this white paper is to help you discover a better way of working with *your business data.*

Traditional Databases

Traditional database tools force designers and users to work with several interfaces that either show the user too much data or don't show enough. The predominant interface uses square tables of columns and rows that present more data than the user needs, duplicating values and obscuring the useful information. For example, consider the following simple table:

Bats & Balls		
Object	**Color**	**Size**
Ball	**Blue**	**Big**
Ball	**Green**	**Big**
Ball	**Green**	**Small**
Ball	**Red**	**Small**
Bat	**Black**	**Big**
Bat	**Blue**	**Big**
Bat	**Green**	**Small**
Bat	**Red**	**Big**

How many colors are there for bats and balls? What is the most frequently used color? Are there any small blue objects? How many blue balls are there? What are the colors of the small objects? What colors are not available for the small objects? Even with this small sampling of data, the answers to these simple questions are clouded by the redundant values.

Tables of this nature and not designed for easy interpretation of data. They are designed to simplify data input and storage. In order to ask questions of the database, one must use a structured language that the computer can understand or rely on the system designer to create forms and reports that answer the questions. To use an existing form, one would enter search values and produce a report. If the form does not allow searching for required values, or the report does not show the needed answers, the user must either resort to a complex query language or request a modification by the system designer.

Forms and Reports

Forms and reports are designed to shield the user from the volumes of data and the complex data structure in the database. These tools provide static answers to specific questions that were anticipated during the design phase. When a new question surfaces, it usually means the system designer must make modifications to the application, changing reports and forms and possibly the database itself. This is typically a very time consuming process of scheduling, development, testing, debugging, and turnover. The end result is that the new question may not be answered due to time or resource constraints.

By using forms and reports to shield users from databases, we also shield the user from the critical relationships to be found in the underlying data. The relationships among the data elements are essential for properly understanding the data. Tables and forms are great interfaces for entering and updating data within a database, but they inhibit learning and understanding of the accumulated data.

Application Design

Compounding the issues surrounding the interface, application development tools are generally too difficult for end users to create their own applications. These tools are structured so that a database application programmer will design and develop an application, from the data source to the user interface. Application designers must anticipate the needs of the users; properly construct primary keys, indexes, and forms to optimize the application performance for specific uses; and modify applications frequently to satisfy new, unanticipated requests from users. Users have to modify the way they work to fit the resulting application, make frequent requests for changes in the application, and wait through long delays while their requests are fulfilled. With each new application, users typically have to learn a new interface, form, and procedure for properly working with the application.

Is there a better solution?

Concepts

QlikView offers a new perspective for database applications. Contrary to the traditional database tools, QlikView has a unique interface that is not focused on tables or forms or reports. Instead, QlikView gives the user various presentation interfaces to suit the needs of a particular circumstance. Instead of working with inflexible tables, restrictive forms, or complex query languages, the user works with a color-coded, point-and-click interface. Instead of concealing relationships in the underlying data, QlikView brings the relationships to the surface to aid the user in understanding and interpreting the data.

Complement Existing Database Technologies

As already stated, relational databases are good for data entry and update, but they obscure information stored in the database from the user. QlikView's associative database is specially designed to allow users to get maximum benefit from all of the information stored in their databases.

QlikView acts as a user-focused complement to your existing databases. Data entry and update must still be performed against a traditional database. QlikView then extracts the data and builds a powerful, associative database that enhances the way users work with the information.

Recall the table presented in the previous section. While QlikView can present tabular data in the same format, it can also be presented in a much more meaningful way. The key to the presentation objects available within QlikView is that the user can work with individual columns or fields from a table without having to deal with the rest of the table.

Associative Database

Most traditional databases are built upon a relational model. Records are broken apart to reduce redundancy, and key fields are used to put the records back together at the time they are used. Database programmers are required to make tradeoffs between increased speed at the cost of more space and more time to add or edit records, and the database user often suffers based on these decisions.

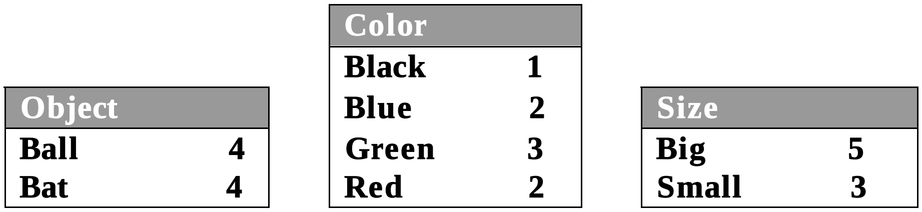
QlikView creates an associative database as it loads data from a data source. While the details of an associative database are beyond the scope of this paper, it is an important distinction that an associative database is different from a relational database. In particular, the associative database operates much faster and requires significantly less space than an equivalent relational database. The associative database is designed to allow the database user maximum flexibility and information when working with the database.

QlikView's associative database optimizes the data as it loads. Redundant field data is removed, and tables are automatically linked together. Indexes are not required, making every field available as a search field without any performance penalty. Also, the space needed for the associative database is only a fraction of the space required for the same data represented in a relational model.

List Box

The basic building block of a QlikView application is the list box. A list box is a movable, resizable object that presents the data from a single column of a table. Rather than listing the duplicate values, only unique values are presented. If desired, the number of occurrences of each distinct value can also be listed.

The following illustration contrasts the tabular presentation of the bat and ball data with the same data presented using QlikView list boxes. The number of occurrences of each field value (frequency) is also shown in each list box. Answering the same questions posed earlier is a simple matter with this interface:



Object	
Ball	**4**
Bat	**4**

Color	
Black	**1**
Blue	**2**
Green	**3**
Red	**2**

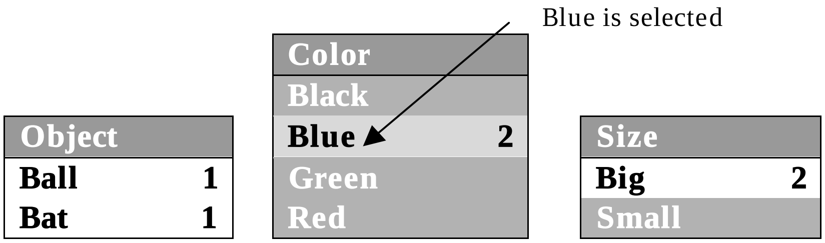
Size	
Big	**5**
Small	**3**

How many colors are there for bats and balls?

There are four distinct colors.

What is the most frequently used color?

Green occurs three times in the table.

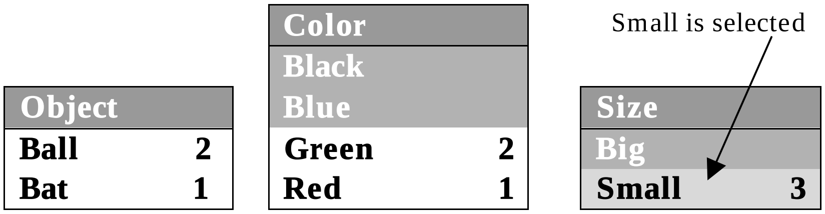


Are there any small blue objects?

Blue is not associated with Small.

How many blue balls are there?

Ball occurs one time for the color Blue.



What are the colors of the small objects?

Green and Red are associated with Small.

What colors are not available for the small objects?

Black and Blue are not associated with Small.

As you can see from these examples, asking and answering questions is a simple matter of point and click. In the image directly above, the value *Small* has been selected (notice the highlighting), and the white values in the other list boxes reflect the objects and colors associated with size *Small*. The numbers to the right of each value represent the number of times the value occurs in relation to the current selection.

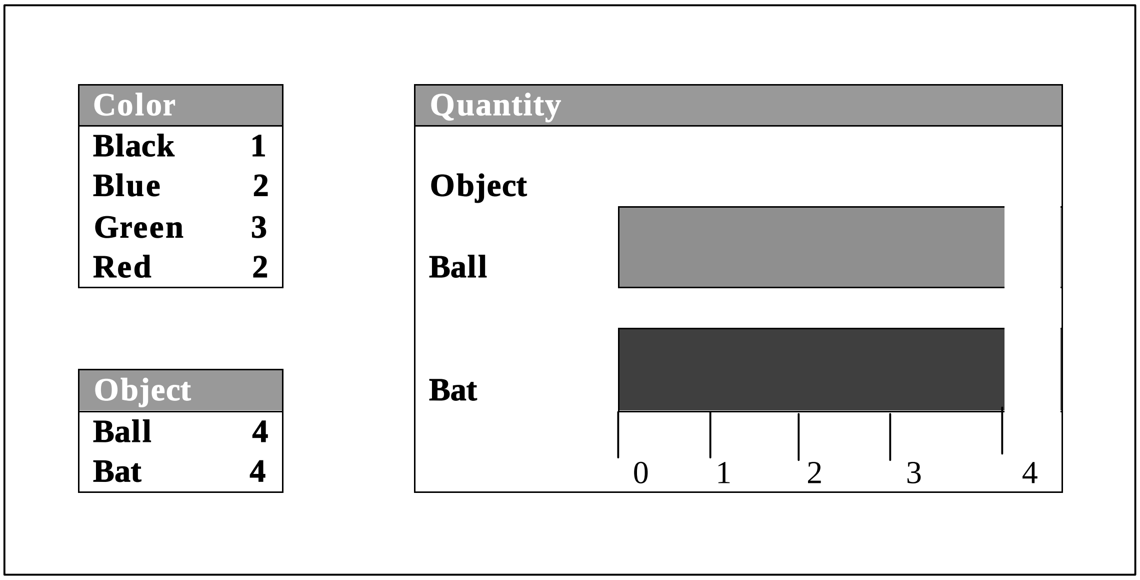
Point-and-click Queries

What is a query? A query is simply a question. Typically, you know some piece of information, and you want to know more about it. For example, what if you wanted to know the address for QlikTech? You know two pieces of information: the company you want to know more about is *QlikTech*, and the item you want to know is *Address.*

The user forms a query in QlikView simply by clicking the mouse on a field value or other item of interest. In a list box, the user clicks on one or more values of interest to select them. QlikView immediately responds to the mouse click and updates all objects displayed on the current sheet.

QlikView uses colors to allow the user to interpret the results of a query (a mouse click). The values that the user actually clicks are highlighted in green and are called *selected* values. The values that are associated with the clicked values remain white and are called *optional* values. Any values that are not associated with the clicked values are called *excluded* values and are dimmed with a gray background.

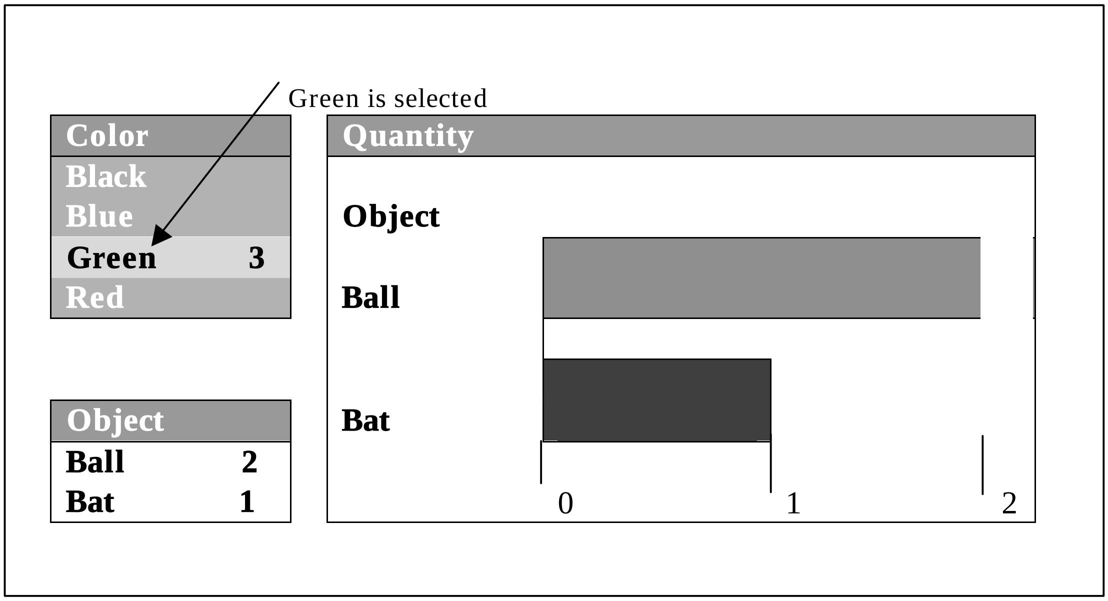
Consider the following example. Three objects are displayed: two list boxes with several values and a bar chart. The list boxes indicate the various objects available and the colors the objects come in. The bar chart shows the quantity of each object.



Color	
Black	1
Blue	2
Green	3
Red	2

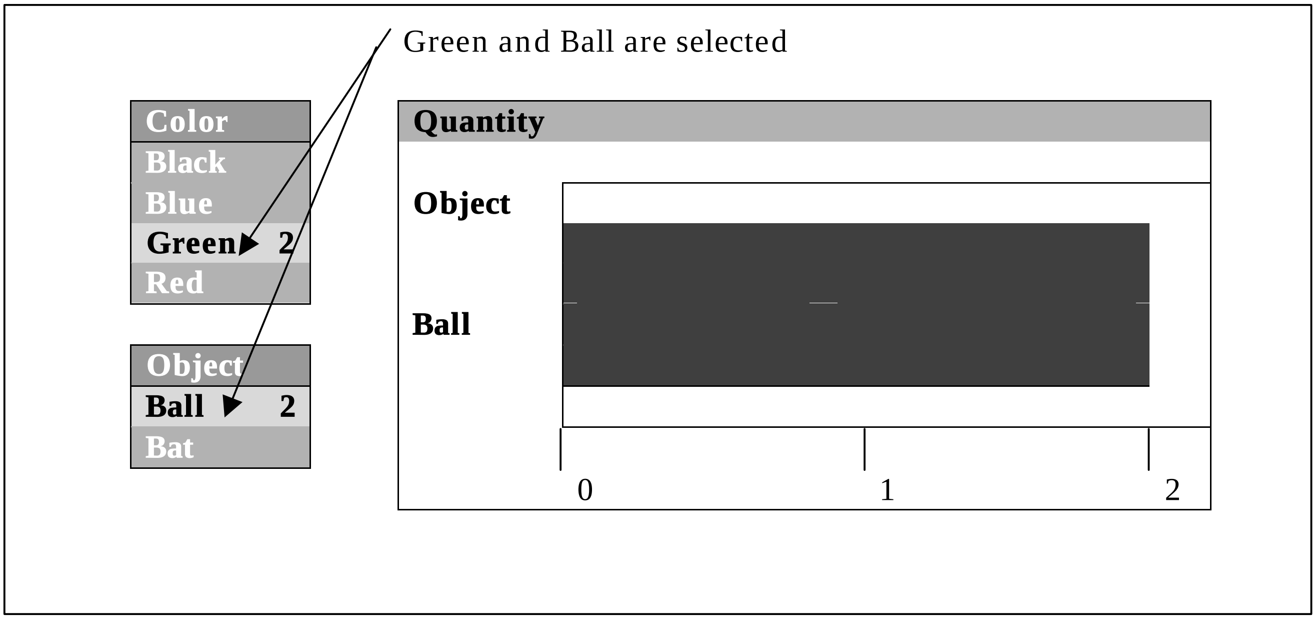
Object	
Ball	4
Bat	4

In order to issue a query, simply click one or more values. For example, to find out the quantity of all Green objects, click on the value *Green* in the list box. Below, you see the result of the selection.



The value *Green* in the Color list box is highlighted to indicate that it is the selected value. Blue, Black, and Red in the Color list box turn gray to indicate that they are not associated with Green, which means there are no Green objects that are also Blue, Black, or Red. Also, notice that the chart has updated the bars for only those objects associated with the value Green. The quantities are also updated to reflect the quantities of Green Balls and Bats.

QlikView allows selections in this manner in as many objects as needed. We could continue to refine the query by clicking on the value Ball in the Object list box (refer to the example below). QlikView would respond by highlighting the value Ball to indicate that it is also selected (in addition to the value Green in the Color list box). The value Bat turns gray indicating that it is not associated with Green and Ball, and the chart is also updated to show only quantities for Green Balls.



Point-and-click queries are not limited to list boxes. In QlikView, you can click in just about any object to make a selection. In order to select the value Ball, we could have clicked the bar in the chart belonging to Ball, or the value ball in the list box to produce the same result. The value of Ball would turn green in the Object list box, and the chart would be updated to reflect that one only one object, Ball, is currently selected.

Multiple Sort Options

Since each field of data can be displayed in its own list box, it makes sense that you would want to sort each list independently of all others. When you are scrolling through a list box, you want the values to appear in some sorted order appropriate to that field. QlikView allows you to sort each list box independently and according to multiple sort specifications.

One or more of the following algorithms can apply to each list box in either ascending or descending order. In addition, the same field (column) can be displayed in multiple list boxes - each with its own sort order.

- State: selected and optional values can be sorted to the top or bottom of the list box
- Frequency: values are sorted by frequency of occurrence
- Numeric Value: values are sorted according to their numeric value
- Text: values are sorted alphabetically
- Load Order: values are sorted according to the way they occurred in the original source database

Sorting by State is a very important feature of QlikView that enables users to quickly interpret the results of a query. When the number of values in a list box is more than can be displayed, scroll bars are added to the list box. This could make it difficult to find the selected or optional values in the list without having to scroll through the entire list box. Enabling the sort by state option for a list box causes all selected (green) or optional (white) values to sort to the top of the list box so the user can easily see them.

Powerful Searching

Remember that a list box represents the distinct values found in a specific field in one or more tables. By displaying only distinct values, the number of values presented is reduced to a minimum. However, the number of values listed can still be hundreds of thousands or more. Finding a specific value could be difficult if you had to scroll through the complete list.

Fortunately, QlikView allows you to search through the list as simply and quickly as typing on the keyboard. When you start typing, QlikView immediately begins searching through the current list box to find values matching your criteria. Single character and multi-character wildcards are supported, as well as greater than and less than symbols to enable searching for numeric and date ranges.

As QlikView locates values matching the criteria typed so far, those values are pulled together to the top of the list box and colored white. Values that do not match the search criteria turn gray and are grouped together at the bottom of the list box. Once the specific value or values you are searching for come into view, you can either click on the values with the mouse to select them or press the enter key to select all marching values.

Personalized Applications

Another exciting feature of QlikView is that each user can personalize an application to meet his or her specific needs. While it may be necessary to have an application designer build the original application and define the data that will be available in QlikView, any user can quickly and easily add or move sheet objects to suit the way he or she works.

All objects displayed on a sheet are automatically associated. When a user adds a new sheet object, as soon as the object is displayed the user can begin to use the object to make selections in the database.

For example, a user may wish to add a new table box to an existing application. All that is required is that the user specifies the fields to be included in the table box. Once that is done, the table box will appear on the current sheet and the user can begin to use it.

In addition to adding sheet objects, users can move objects, resize objects, remove objects, add new sheets, rearrange sheets, and remove sheets. Of course, the system designer can restrict a user from performing these functions, but generally users are allowed to personalize an application to fit their specific working practices.

Key Features

QlikView comes with many features for performing data analysis, navigation, and presentation. The following pages discuss some of the key features that set QlikView apart from the rest of the database world.

System Requirements

QlikView runs on any IBM PC compatible computer running Windows 95, Windows NT 4.0, or a later version. While and 80386DX processor is all that is required, QlikView takes full advantage of the system hardware, so faster processors directly affect application performance.

Available system RAM determines the maximum size of a database application. Having more RAM available allows larger applications to be created, integrating data from more sources. 16MB or more of RAM is recommended for most applications.

Data Capacity

QlikView is designed to handle a remarkable amount of data. There is no limit to the number of tables allowed in an application. In addition, there is no limit to the number of fields, rows, or cells in a single table. The only limit imposed by QlikView is that a single field cannot contain more than 2,000,000,000 (that's two billion) distinct values. Beyond that, RAM is the only other factor that limits the size of an application.

Since QlikView works with data in memory, it is best suited for applications with manageable data sizes. Typical applications will be under 100MB of data in a relational database, resulting in a 10MB - 20MB QlikView database.

The maximum size of QlikView application is closely tied to the available RAM on the system where the application will run. However, it is not as easy as looking at the size of a relational database and comparing that to the RAM on the system to determine if the application is appropriate for QlikView. As QlikView loads data from a source database, the data is highly optimized, usually resulting in a QlikView application of only 10-30% of the size of the original source. This means that for systems with 64MB of RAM, a typical 256MB relational database could be loaded into QlikView.

Data Sources

The design idea behind QlikView is to make a tool that users can use to present, navigate and analyze data in a fashion that enhances the way the way they work. To this end, QlikView works in cooperation with other tools that are used to create and store data. QlikView users use a database or spreadsheet to store their data and use existing tools or applications for data entry. QlikView simply provides a consistent, integrated access point to work with the data from those data stores.

QlikView loads data from the following data sources:

- Delimited text files with any single character delimiter
- Fixed-record length text files with fixed-width fields
- AS/400 Data Interchange Format (DIF)
- Microsoft Excel spreadsheets
- QlikView's associative database files
- ODBC enabled data sources
- OLEDB enabled data sources

QlikView includes the ability to load from these sources (except for ODBC/OLEDB) using HTTP or FTP. This means that the file source may reside on a web server or FTP server at a remote location, and QlikView can connect to the server and load the data source directly.

Load Script

In order to pull data from a data source, QlikView executes a load script. The load script defines the source databases and tables and fields that should be loaded into QlikView. In addition, you can calculate new variables and records using over 99 functions available in the script.

In order to help you create a load script, QlikView includes a wizard that will generate the script. For simple applications, you simply need to point at the file you want to load; QlikView will automatically determine the type of file and generate the script required to load all of the data from the source. For more complex applications, it may be necessary to manually edit the script to load or combine sources in a non-standard way.

Once the load script is generated, pushing the run button causes the script to execute. Within a few seconds, QlikView will extract the data from the source according to the instructions in the script and make the data available for use. The data can be reloaded at any time by simply pushing the reload button on the toolbar. Partial reloads are also supported to replace only specified tables or append new data to an existing table.

QlikView can load data that is stored in a variety of formats. Data can be loaded from generic tables, cross tables, mapping tables (data cleansing), and interval-matching tables. Tables can be joined, concatenated, sampled, and linked to external information such as other programs, bitmaps, URLs, etc.

Associate Query Logic

QlikView is based upon patented technology called Associative Query Logic (AQL). AQL enables QlikView to work with millions of cells and still respond to queries in less than a second. High-speed associations occur as the user clicks in the various sheet objects, and the display is updated immediately.

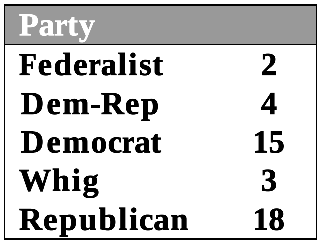
By replacing resource intensive relational joins with AQL, QlikView removes the need to pre-aggregate data for most applications. In Online Analytical Processing (OLAP) applications, QlikView removes the need to define complex dimensional hierarchies and generate cubes. QlikView performs calculations on the fly, allowing any user to become a sophisticated knowledge worker.

Multiple Sheet Objects

In a previous section, we looked at the list box. The list box is the most basic object available in QlikView. There are, however, many other objects available. Combining various objects together on a sheet creates a powerful application for working with and analyzing any type of data.

List Box

As described earlier, a list box is a movable, resizable object that presents the data from a single column of a table. Rather than listing the duplicate values, only unique values are presented. If desired, the number of occurrences of each distinct value can also be listed, and the list box can be set to show multiple columns. If the same field occurs in multiple tables, all of the values from all of the fields of the same name are combined into the single list box.

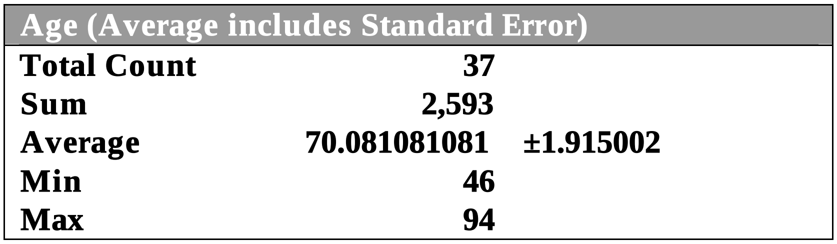


Party	
Federalist	2
Dem-Rep	4
Democrat	15
Whig	3
Republican	18

List boxes are extremely fast and very powerful presentation objects. With a couple of mouse clicks, you can add frequency information about each value to the list box, change the sort order of the list box independent of any other object, allow multiple column display, or change many other properties to control exactly how the list box looks and behaves.

Statistics Box

The statistic box allows you to choose from up to 15 different statistics to display about a particular field. Statistics such as Min, Max, Count, Sum, Standard Deviation, and Standard Error can be continually displayed for any field. Many of the statistics allow clicking on the value in the statistics box to select the values contributing to the statistic in the corresponding list box. For example, clicking on the Min value in a statistics box will select the minimum value in the corresponding list box.



Age (Average includes Standard Error)		
Total Count	37	
Sum	2,593	
Average	70.081081081	±1.915002
Min	46	
Max	94	

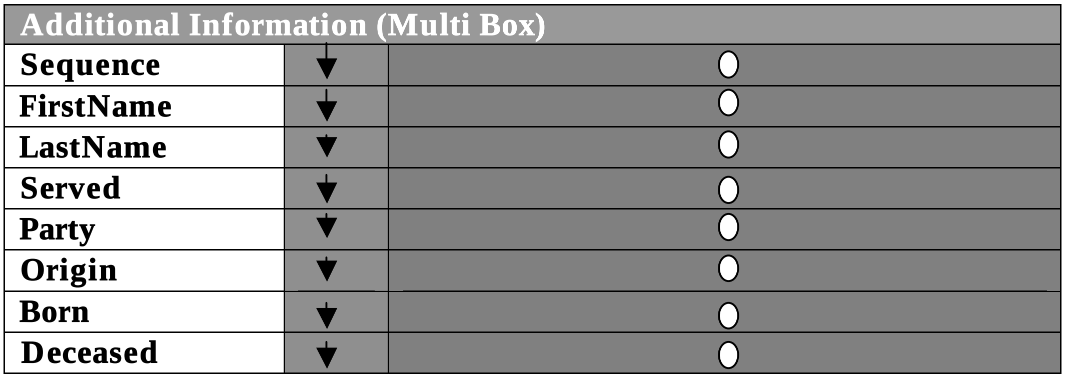
Multi Box

The multi box appears to be nothing more than a combination of drop-down list boxes; however, its true capabilities go much deeper. As a basic object, it allows one or more fields to be displayed as a group of stacked, drop-down lists. These drop-down lists have most of the same functionality of the list box without taking up as much space on the screen. When a list is opened, the user can search and select multiple values just like in the list box.

Multi boxes use colored beacons to indicate their contents. If a single value is selected or optional (green or white, respectively), that value is displayed when the list is closed. If multiple values are selected or optional, a green or white beacon, or dot, is displayed when the list is closed. When a list is empty or contains only excluded values, no beacon is displayed. Using the visual indicators, it is very easy to interpret the results of a query and to see fields that could use further investigation.

An aspect of the multi box is its ability to sort fields as well as values. In the same way that a list box can be set to float selected and optional values to the top of the list, the multi box can be set to float available fields to the top.

A multi box could contain hundreds or thousands of fields. When there are more fields that can fit the specified display, QlikView automatically adds scroll bars to the multi box. When a selection is made in the application, those fields that have values associated with the selected value will float to the top of the multi box. This can be a very powerful tool in designing applications with many fields that do not apply to every record.



Additional Information (Multi Box)		
Sequence	▼	○
FirstName	▼	○
LastName	▼	○
Served	▼	○
Party	▼	○
Origin	▼	○
Born	▼	○
Deceased	▼	○

Table Box

The table box is probably the most familiar object. This is similar to a spreadsheet in most other applications. A table box is used to present logical combinations of multiple fields. Data can be combined from multiple data sources and displayed in a single table.

Party	Origin	Name
Federalist	Massachusetts	John Adams
Federalist	Virginia	George Washington
Whig	New York	Millard Fillmore
Whig	Virginia	Williams Henry Harrison
Whig	Virginia	Zachary Taylor

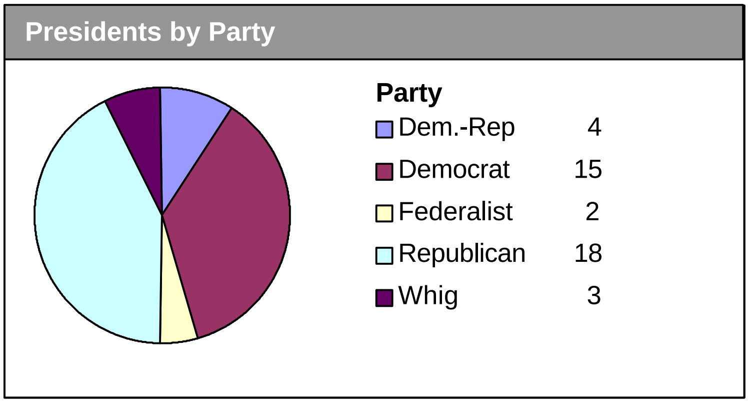
Tables require displaying duplicate values. When the data demands that the user see the values side by side, tables are often the best way to present it. The user can still make selections by clicking any value in a table box and can sort the table by any field.

Table boxes are also convenient mechanisms for exporting or printing values from a QlikView application. Table boxes can be exported to a delimited text file for processing by another application or can be printed to create formatted reports. Table boxes also have several defined styles to enhance their appearance in the application.

Charts

QlikView provides an assortment of charts. Select from any of the following formats:

- Bar charts
- Line charts
- Pie charts
- Scatter charts
- Combo charts

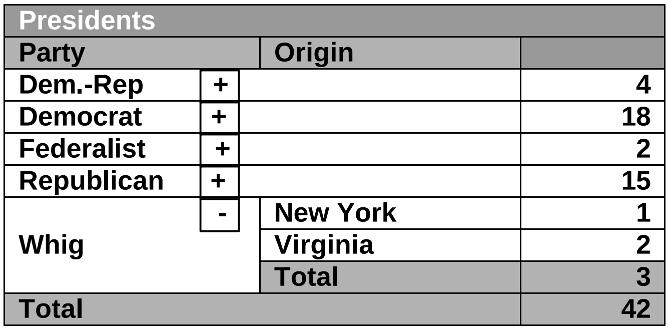


Each chart type has a host of options to choose from to set properties like colors, axes, vertical or horizontal orientation, numbers on bars, vertical text, etc. Values within charts can be selected by clicking on a specific bar, line or slice, or by performing a click-and-drag operation to select multiple values. Clicking values in the chart legend will even make a selection.

Charts can be defined with multiple expressions on a single chart or with a button to cycle between the defined expressions. Each chart supports an unlimited number of dimensions and expressions. You can add trend lines to the charts using any of six standard formulas, and even add forecasts and backcasts to expand on the existing data.

Pivot Table

QlikTech has a patent pending on the pivot table added to QlikView 4.0. This powerful new object supports an unlimited number of dimensions and expressions. You can pivot variables and expressions using drag-and-drop to move them between the horizontal and vertical axes.



Presidents

Party		Origin	
Dem.-Rep	+		4
Democrat	+		18
Federalist	+		2
Republican	+		15
Whig	-	New York	1
		Virginia	2
		Total	3
Total			42

Dimensions can be selectively expanded or collapsed as appropriate to get the exact view of the data required.

Visual cues have been added to the pivot table to allow user-defined color-coding of cell values within a specific range. The ranges can be specified as constants or using dynamic expressions to provide clear data interpretation.

Straight Table

The straight table is a cross between the pivot table and the table box. While the table box cannot include expressions, the straight table displays both variables and expressions. Pivot tables can display partial sums and sub-totals, where the straight table does not. Straight tables can be sorted by any column, and visual cues are supported.

Button

Create buttons and place them on sheets to control various aspects of the application. Buttons can be made from custom bitmaps or text or both. Pressing a button can trigger any of a number of actions including launching custom macros.

Text

Text objects allow you to customize the look of an application. Free-floating text can be placed anywhere on a sheet and can be designed with custom backgrounds, fonts, and colors. Text objects can even include custom expressions allowing the text to change based on currently selected values in the database.

Sheets

Each of the above objects is placed on a sheet in QlikView. An application can have as many sheets as required to organize and show the required information. Each sheet has a tab with a name to identify it from the other sheets and to allow for easy navigation from sheet to sheet. Sheets supply the background and wallpaper for the application and serve to group specific tasks together.

Expressions

QlikView allows expressions to be used in a variety of ways. Expressions can be simple, invoking a single function, or very complex, combining multiple functions, operators, and constants. Use expressions in text objects located anywhere in the application: charts, pivot tables, straight tables, titles, or anywhere a dynamic value needs to be displayed.

QlikView expressions can combine data from different fields using over 90 functions. Functions include categories like aggregation functions, mathematical functions, data and time functions, string functions, and conditional functions. Almost all of the functions available during the initial load script, when the data is extracted from the source database, are available at runtime for dynamic calculation.

Variable Groups and Drill Down

QlikView charts support another flexible feature called variable groups. Groups are lists of variables that are defined as cyclic or drill-down. These groups can flow naturally, e.g., Year, Quarter, Month, Day, or mix any type of variables, e.g., Year, Customer, Product. When a group is used as the variable specification in a chart, the chart gains new functionality depending on the type of group used.

Drill-down groups specify variables that display sequentially as you drill-down in the chart. When only a single value (one bar, one slice, or one point, depending on the chart type) is available to be displayed in the chart, QlikView automatically drills down to the next variable level. For example, a bar chart could be displayed showing total sales for 12 months. When a single month's bar is selected, the chart can automatically drill down to display sales for the 4 weeks in that month. When a single week is selected, the sales for the 7 days of the week are displayed. When a single day is selected, the sales for the individual customers could be displayed.

Traditional OLAP tools must pre-calculate aggregations, so their drill-down capabilities are limited to moving up and down a defined hierarchy. QlikView performs calculations on the fly, which means drill-down groups are not limited to a simple hierarchy. Instead, QlikView allows any variables to be grouped together as appropriate for the application, and the variables within the group can be changed with just a few mouse clicks without requiring modifications to the chart definition.

Alternatively, groups can be defined as cyclic instead of drill-down. When a cyclic group is specified as the variable in a chart, a cycle button is automatically affixed to the chart. When the cycle button is pressed, the chart displays the next variable in the group. Alternatively, the user can right-click on the button to see a list of variables or expressions in the cycle and move directly to any one. In the example above, the chart would still start out displaying total sales for 12 months. When the cycle button is pressed, the chart would change to show the sales for all 52 weeks of the 12 months (or any months that were selected.) The chart would cycle through each of the variables with each press of the cycle button.

Linking External Data or Info

QlikView provides a versatile method for linking external data with field values in a QlikView application. The associated external data can be displayed automatically when the value is selected or manually via an Info button. When the external data item is a file name, QlikView will launch the file within its own application. When the external data is a BMP or a JPG graphic image, or any text that is not a file name, QlikView will display the data using its internal viewer.

Info associations allow pictures to be associated with individual names, descriptive text to pop up explaining a particular field value, and application launching with specific parameters based on a QlikView selection.

Layout Design

QlikView includes a wealth of features to help you lay out the data presentation in exactly the format desired. All of the expected features are available, like drag-and-drop editing to move or copy objects within a sheet or between sheets. Multiple objects can be selected and then copied, moved, aligned, or spaced as a group. Font and border options can also be specified for a group of objects at the same time. In addition, objects can be placed on one of three layers to control their appearance in relation to other objects.

The appearance of each object can also be controlled. With just a click or two, designers can choose the font, alignment, and borders for each object. In addition, double-clicking an object's title bar minimizes it, allowing better use of the sheet space. When an object is not in use, it can be minimized on the sheet and reopened at a later time.

Text objects are a convenient way to enhance the look of an application. Text objects can be used simply to provide a color area on a section of a sheet, or to display text. While the text object can be loaded with static text, QlikView's powerful expression engine is also available within text objects to allow dynamic text and numbers anywhere within the application.

The appearance of each sheet can be customized to provide a rich look for an application. Choices range from a single background color to tiled or centered wallpaper. Combined with the flexibility of the text objects, sheet backgrounds create a very professional appearance for any application.

Visual Basic Script Support

The QlikView automation interface offers application designers the ability to build complete, seamless applications based on the QlikView user interface. QlikView includes a complete automation facility based upon Visual Basic Script (VBScript). With this powerful object-oriented programming language, it is possible for QlikView application designers to gain access to most of the objects and properties within an application.

Programmers can develop VBScript macros to add specific functionality to an application. Macros can be attached to button objects that a user must click to activate, or the macros can be attached to various QlikView events. For example, a macro can be automatically invoked whenever an application is opened, when the load script is executed, or when a selection is made in a list box.

VBScript macros can also be used to add functionality to the load script execution. VBScript functions can be invoked within the load script to perform specific processing, allowing access to each record as it is read from the source database.

In addition to VBScript macros within QlikView, the Automation interface also exposes the QlikView application objects to external programs. Programmers can write programs in any language that supports Automation, like Visual basic or C++, which manipulate the Qlikview application.

Security

QlikView can control access to an application. When an application requires secure access levels, adding a security database to the application is a simple matter. The security database defines the users of an application and the type of access that each user has. Access levels are broken into three groups: ADMIN, USER, and NONE. ADMIN level users have complete access to all areas of an application. USER level users have restricted access. All other users are not allowed to use the application.

The security database is loaded into QlikView in similar fashion to normal application data. The security database can consist of any number of associated tables. Within the database, QlikView recognizes four field names that control each user's access level: USERID, PASSWORD, ACCESS, and SERIAL. Other fields can be added to create associations or user groups as needed, and any of these four fields that are not needed for a particular security profile can be omitted.

For example, if the PASSWORD field is not included in the security database, a user supplying a valid USERID is allowed access. If both USERID and PASSWORD are included in the security database, then the user must supply both a correct USERID and PASSWORD to be granted access to the application. For added security, access can also be restricted by a full or partial QlikView serial number.

The USER access level is allowed restricted access to the QlikView application. The application designer can specify system attributes that should or should not be restricted for a particular application. For example, users may be allowed to add their own sheets and objects to an application but not remove or alter sheets created by the designer.

Making an Application

Simple applications can be created within just a few minutes using QlikView's wizards. More complex applications integrating data from various sources and displaying trend analysis charts and pivot tables may take a little bit longer. The best way to understand how simple it is to create and use a QlikView application is to step through the process involved.

Locate the Data Source

The first step in creating an application in QlikView is to determine what data you wish to load. While it is possible to include inline data in the QlikView load script, application data will almost always come from an existing file, spreadsheet or database. You may load data from a single source file or database, or you may load and integrate data from many different sources at the same time.

The source file should be arranged with each record of the file containing one record of data. The first row should contain field labels, although you can always choose to change the labels in the wizard or in the script. If the data will come from a text file, each file will typically be treated as a single table. When working with spreadsheets, each tabbed sheet will be treated as a table.

If you load data from multiple files or tables, QlikView will automatically associate fields with the same name. If tables should be associated and the association fields are not named the same, you need to rename the fields in the load script. If fields in different tables have the same name but should not be associated, you need to rename those fields in the load script to break the association.

Create the Load Script

Once the data has been determined, a load script must be created to copy the data from the data source into QlikView's associative database. Creating the load script is simplified by the use of wizards that construct script statements for the supported file types.

If the source data is contained in text file or Microsoft Excel spreadsheet, the typed file wizard will guide you through a few simple steps to create load script statements. For ODBC databases, two wizards will help you create a connection to the database and build an SQL Select statement to read the database. You will use the wizards for each table that must be read. With just a few dialogs, you will specify the file name, the delimiter, the fields to read, and the new names of any fields. After completing the specification for a table or file, the wizard will add a load statement to the script.

Execute the Load Script

After the load script is complete, the script must be executed by pressing the Reload button on the tool bar or selecting the Reload option on the File menu. During the load process, QlikView examines each statement in the load script and processes them in sequential order. At the completion of the load script, a copy of all of the data referenced in the load script is loaded and available in the QlikView application.

Place Objects on a Sheet

In order to use the data in the QlikView application, you must place list boxes or other objects on one or more sheets. The actual objects that should be used and how they should be grouped into sheets depends on the specific application.

Typically, you should start by placing the main fields as list boxes. Using the Select Fields dialog box, one or more fields can be quickly added to the current sheet using the default list box format. Once the list boxes are created, their individual properties can be adjusted to add frequency counts, change fonts, change borders, etc. They can also be rearranged on the sheet by simply dragging each list box to its desired location.

Any field loaded by the script can be displayed on the same sheet. If fields were loaded from different tables, they will be automatically associated together and respond to clicks as if all fields belong to one large table.

Start Using the Application

As soon as the first object is created on a sheet, the application is available for use. All objects are automatically associated together, and clicking in any object initiates a query.

Add More Sheets and Objects As Required

Finally, continue to add and arrange objects on sheets until the application achieves the functionality desired. It may be necessary to add more customization to the load script by taking advantage of the functions available or to write VBScript macros to automate certain actions.

Objects can be copied or moved to other sheets by dragging the object and dropping it on the sheet tab of another sheet. Layout controls on the view menu allow easy alignment and spacing of sheet objects to help create an orderly appearance. In addition, wallpaper and text objects can be placed on the sheet to give the application a polished and professional look.

Within just a few minutes, a complete working application can be created. The application can be distributed to other users who can add customization to the layout without any need to access the original data source. For example, users can add their own sheets, pivot tables and charts (or any other object) to make the application fit the way they work.

Conclusion

QlikView is an exciting complement to your existing databases. For the application designer, QlikView brings together rapids application design capabilities and a logical, consistent interface. For the application user, QlikView offers very high-speed query and calculations along with the ability to personalize an application to the way you work. Whether you are working with complex OLAP applications or simple data navigation, QlikView is the one tool for every user.

QlikView allows the application designer to focus on specifying the data universe for an application. Utilizing QlikView's powerful scripting language, the application designer can load, combine, aggregate, and manipulate data from many different data sources into a single seamless application.

Once the data is loaded, the layout design can be accomplished by the system designer alone or in cooperation with the end user. In fact, the end user is able to customize the application, if allowed, by adding or rearranging sheets and object to suit the way he works. The consistent interface, no matter what the application, enhances the user's experience and productivity when working with data.