



# QlikView

**tutorialspoint**  
SIMPLY EASY LEARNING

[www.tutorialspoint.com](http://www.tutorialspoint.com)



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

## About the Tutorial

---

QlikView is a leading Business Discovery Platform. It is very powerful in visually analyzing the relationships between data. It does in-memory data processing and stores the data in the report itself that it creates. It can read data from numerous sources including files and relational databases.

It is used by businesses to get deeper insight by doing advanced analytics on the data they have. It even does data integration by combining data from various sources into one QlikView analysis document. QlikView is a leading Business Intelligence and Analytics Platform in [Gartner Magic Quadrant](#).

## Audience

---

This tutorial is designed for all those readers who want to create, read, write, and modify Business Intelligence Reports using QlikView. In addition, it will also be quite useful for those readers who would like to become a Data Analyst or Data Scientist.

## Prerequisites

---

Before proceeding with this tutorial, you should have a basic understanding of Computer Programming terminologies. A basic understanding of any of the programming languages will help you in understanding the QlikView programming concepts. Familiarity with SQL will help you learn it very fast.

## Copyright & Disclaimer

---

© Copyright 2017 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute, or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at [contact@tutorialspoint.com](mailto:contact@tutorialspoint.com)

## Table of Contents

---

About the Tutorial .....	i
Audience.....	i
Prerequisites.....	i
Copyright & Disclaimer .....	i
Table of Contents .....	ii
<b>1. QlikView – Overview.....</b>	<b>1</b>
<b>2. QlikView – Installation .....</b>	<b>2</b>
<b>3. QlikView – Architecture .....</b>	<b>10</b>
Architectural Overview.....	10
<b>4. QlikView – Navigation.....</b>	<b>12</b>
Getting Started Screen .....	12
Menu Commands .....	13
<b>QLIKVIEW DATA LOADING.....</b>	<b>22</b>
<b>5. QlikView – Excel Files.....</b>	<b>23</b>
<b>6. QlikView – Delimited File .....</b>	<b>29</b>
<b>7. QlikView – XML File .....</b>	<b>35</b>
<b>8. QlikView – Web File .....</b>	<b>41</b>
<b>9. QlikView – Database Connection .....</b>	<b>49</b>
<b>10. QlikView – Inline Data.....</b>	<b>56</b>
<b>11. QlikView – Resident Load .....</b>	<b>60</b>
<b>12. QlikView – Preceding Load.....</b>	<b>62</b>
<b>13. QlikView – Incremental Load .....</b>	<b>64</b>
<b>14. QlikView – Data files (QVD) .....</b>	<b>71</b>
Advantages of Using QVD Files.....	71
Creating QVD Files.....	71
Using QVD Files.....	73
QVD File Loader Script .....	73
Displaying Data from QVD File .....	74
<b>QLIKVIEW REPORT INTERFACE .....</b>	<b>76</b>
<b>15. QlikView – Sheet and Objects .....</b>	<b>77</b>
Sheet Properties .....	77
Creating Sheet Objects .....	79
Using Sheet Objects.....	80

Association between Sheet Objects .....	83
<b>16. QlikView – Scripting.....</b>	<b>84</b>
Script Editor Features .....	85
<b>17. QlikView – Functions .....</b>	<b>86</b>
Create Table Box.....	86
List of Functions.....	88
<b>18. QlikView – IntervalMatch .....</b>	<b>90</b>
Creating the Script .....	91
<b>19. QlikView – Aggregate Functions.....</b>	<b>94</b>
Creating the Load Script .....	95
Applying SUM() function .....	97
Applying AVG() function .....	98
Applying MAX() & MIN() function .....	99
<b>20. QlikView – Match Function .....</b>	<b>101</b>
Load Script with Match() Function .....	102
Load Script with Wildmatch() Function .....	104
<b>21. QlikView – Rank Function .....</b>	<b>106</b>
Input Data.....	106
Load Script .....	107
Creating Chart with Rank() Function .....	108
Using Aggregate Function with Rank.....	111
<b>22. QlikView – Peek Function .....</b>	<b>113</b>
Input Data.....	113
Load Script .....	113
Using peek() Function in Calculations .....	115
<b>23. QlikView – RangeSum Function .....</b>	<b>117</b>
Input Data.....	117
Load Script .....	117
Applying RangeSum() Function .....	118
<b>24. QlikView – Documents.....</b>	<b>119</b>
Document Properties .....	119
Setting Document Background Image .....	120
Sheet Objects.....	122
Scheduling a Document.....	123
<b>25. QlikView – List Box.....</b>	<b>125</b>
Input Data.....	125
Load Script .....	125
Creating List Box .....	126
List Box Data .....	128
List Box and Association .....	130

<b>26. QlikView – Multi Box .....</b>	<b>131</b>
Input Data.....	131
Load Script.....	131
Creating Multi Box.....	132
Multi Box Data.....	134
Multi Box and Association .....	135
<b>27. QlikView – Text Object.....</b>	<b>136</b>
Input Data.....	136
Load Script.....	136
Creating Table Box.....	137
Creating Text Object.....	139
<b>28. QlikView – Bar Chart.....</b>	<b>143</b>
Input Data.....	143
Load Script.....	143
Creating Table Box (Sheet Object).....	144
Using the Quick Chart Wizard.....	146
<b>29. QlikView – Pie Chart .....</b>	<b>151</b>
Input Data.....	151
Load Script.....	151
Using the Quick Chart Wizard.....	152
<b>30. QlikView – Dashboard.....</b>	<b>157</b>
Input Data.....	157
Load Script.....	157
Choose Matrices.....	158
Adding Chart to Dashboard .....	161
Using the Dashboard .....	166
<b>QLIKVIEW DATA TRANSFORMATION.....</b>	<b>167</b>
<b>31. QlikView – Data Transformation .....</b>	<b>168</b>
Input Data.....	168
Loading the Data .....	169
The Transformation Wizard.....	170
<b>32. QlikView – Fill Function.....</b>	<b>177</b>
Input Data.....	177
Load Script .....	178
<b>33. QlikView – Column Manipulation .....</b>	<b>185</b>
Input Data.....	185
Load Script .....	185
Load Script for Transformed Data .....	188
<b>34. QlikView – Rotating Tables .....</b>	<b>191</b>
Input Data.....	191
Load Script .....	192
Apply Rotate .....	194

<b>QLIKVIEW DATA MODEL.....</b>	<b>197</b>
<b>35. QlikView – Dimensions and Measures .....</b>	<b>198</b>
Input Data.....	198
Load Script.....	198
Using Dimensions and Measures .....	200
<b>36. QlikView – Star Schema .....</b>	<b>204</b>
Input Data.....	204
Load Script.....	205
Star Schema Data Model .....	207
<b>37. QlikView – Synthetic Key .....</b>	<b>208</b>
Input Data.....	208
Load Script.....	209
Data Model.....	210
Impact of Synthetic key .....	210
<b>38. QlikView – Generating Data.....</b>	<b>211</b>
Data Requirement .....	211
<b>QLIKVIEW DATA ANALYSIS .....</b>	<b>213</b>
<b>39. QlikView – Cross Tables .....</b>	<b>214</b>
Input data .....	214
Crosstable Options .....	215
Load Script.....	217
<b>40. QlikView – Straight Tables .....</b>	<b>219</b>
Input Data.....	219
Load Script.....	220
Create Table Box.....	221
<b>41. QlikView – Pivot Tables.....</b>	<b>225</b>
Input Data.....	225
Load Script .....	226
<b>42. QlikView – Set Analysis .....</b>	<b>232</b>
Input Data.....	232
Load Script .....	233
<b>43. QlikView – Joins .....</b>	<b>239</b>
Input Data.....	239
Inner Join .....	240
Left Join .....	241
Right Join .....	242
Outer Join .....	243

<b>44. QlikView – Keeps .....</b>	<b>245</b>
Input Data.....	245
Inner Keep .....	246
Left Keep.....	250
Right Keep .....	251
<b>45. QlikView – Concatenation.....</b>	<b>254</b>
Input Data.....	254
Load Script.....	255
Concatenated Data.....	257
<b>46. QlikView – Master Calendar.....</b>	<b>258</b>
Input Data.....	258
Load Script .....	259
Create Master Calendar .....	261
<b>47. QlikView – Mapping tables .....</b>	<b>264</b>
Input Data.....	264
Load Script .....	265
Create the Mapping Table .....	267
<b>48. QlikView – Circular Reference.....</b>	<b>269</b>
Input Data.....	269
Load Script .....	270
Data Load.....	271
Data Model .....	271
Resolving Circular Reference.....	272
Rectified Data Model.....	273

# 1. QlikView – Overview

QlikView is a leading Business Discovery Platform. It is unique in many ways as compared to the traditional BI platforms. As a data analysis tool, it always maintains the relationship between the data and this relationship can be seen visually using colors. It also shows the data that are not related. It provides both direct and indirect searches by using individual searches in the list boxes.

QlikView's core and patented technology has the feature of in-memory data processing, which gives superfast result to the users. It calculates aggregations on the fly and compresses data to 10% of original size. Neither users nor developers of QlikView applications manage the relationship between data. It is managed automatically.

## Features of QlikView

QlikView has patented technology, which enables it to have many features that are useful in creating advanced reports from multiple data sources quickly. Following is a list of features that makes QlikView very unique.

- **Data Association is maintained automatically** - QlikView automatically recognizes the relationship between each piece of data that is present in a dataset. Users need not preconfigure the relationship between different data entities.
- **Data is held in memory for multiple users, for a super-fast user experience** - The structure, data and calculations of a report are all held in the memory (RAM) of the server.
- **Aggregations are calculated on the fly as needed** - As the data is held in memory, calculations are done on the fly. No need of storing pre-calculated aggregate values.
- **Data is compressed to 10% of its original size** - QlikView heavily uses data dictionary. Only essential bits of data in memory is required for any analysis. Hence, it compresses the original data to a very small size.
- **Visual relationship using colors** - The relationship between data is not shown by arrow or lines but by colors. Selecting a piece of data gives specific colors to the related data and another color to unrelated data.
- **Direct and Indirect searches** - Instead of giving the direct value a user is looking for, they can input some related data and get the exact result because of the data association. Of course, they can also search for a value directly.

## 2. QlikView – Installation

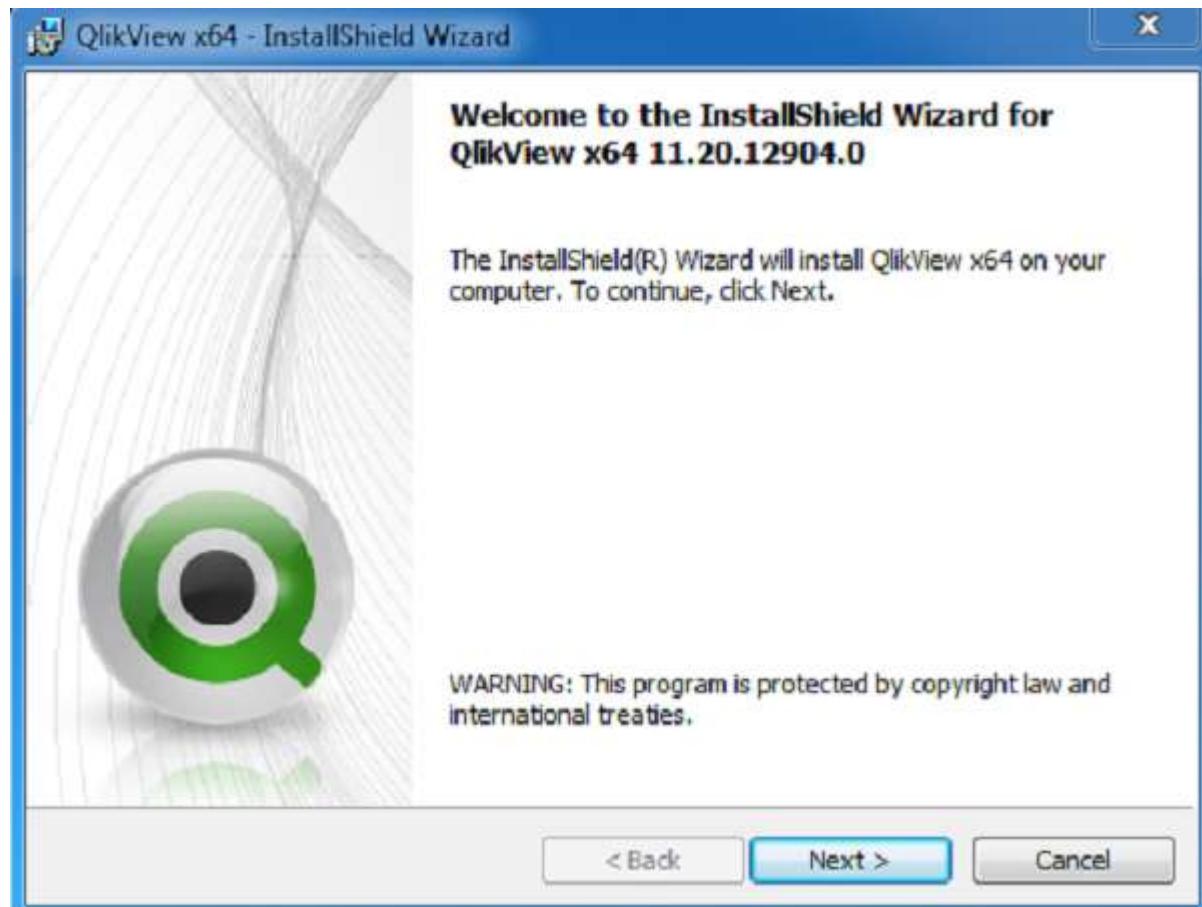
### Download QlikView

The Free Personal Edition of QlikView can be downloaded from [QlikView Personal Edition](#). You need to register with your details to be able to download.

After downloading, the installation is a very straightforward process in which you need to accept the license agreement and provide the target folder for installation. The screen shots given below describe the entire setup process.

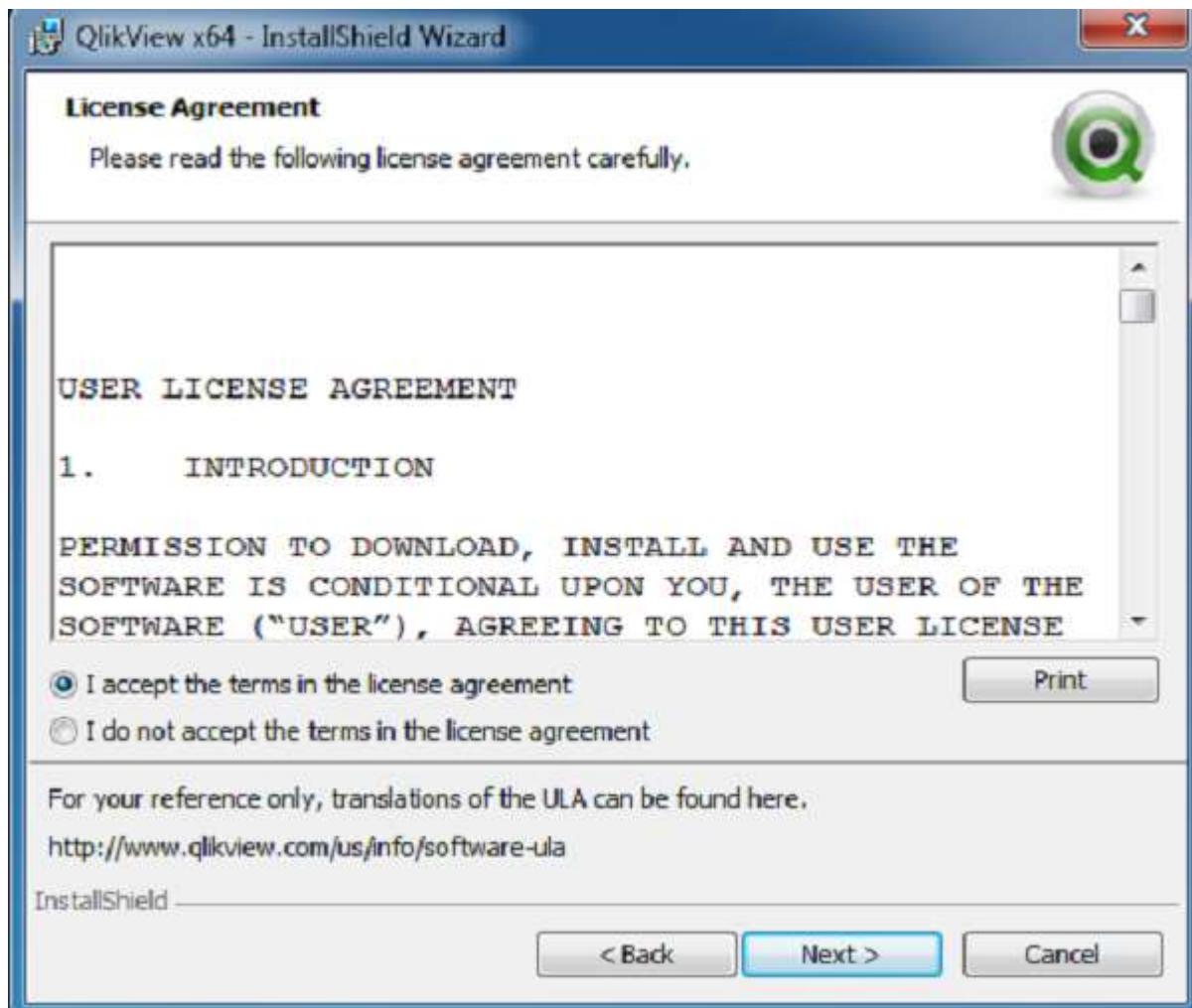
### Start the Installation Wizard

Double clicking the **QlikViewDesktop\_x64Setup.exe** will present a screen to select the language of your choice. On selecting English, the following screen is displayed. Then click **Next**.



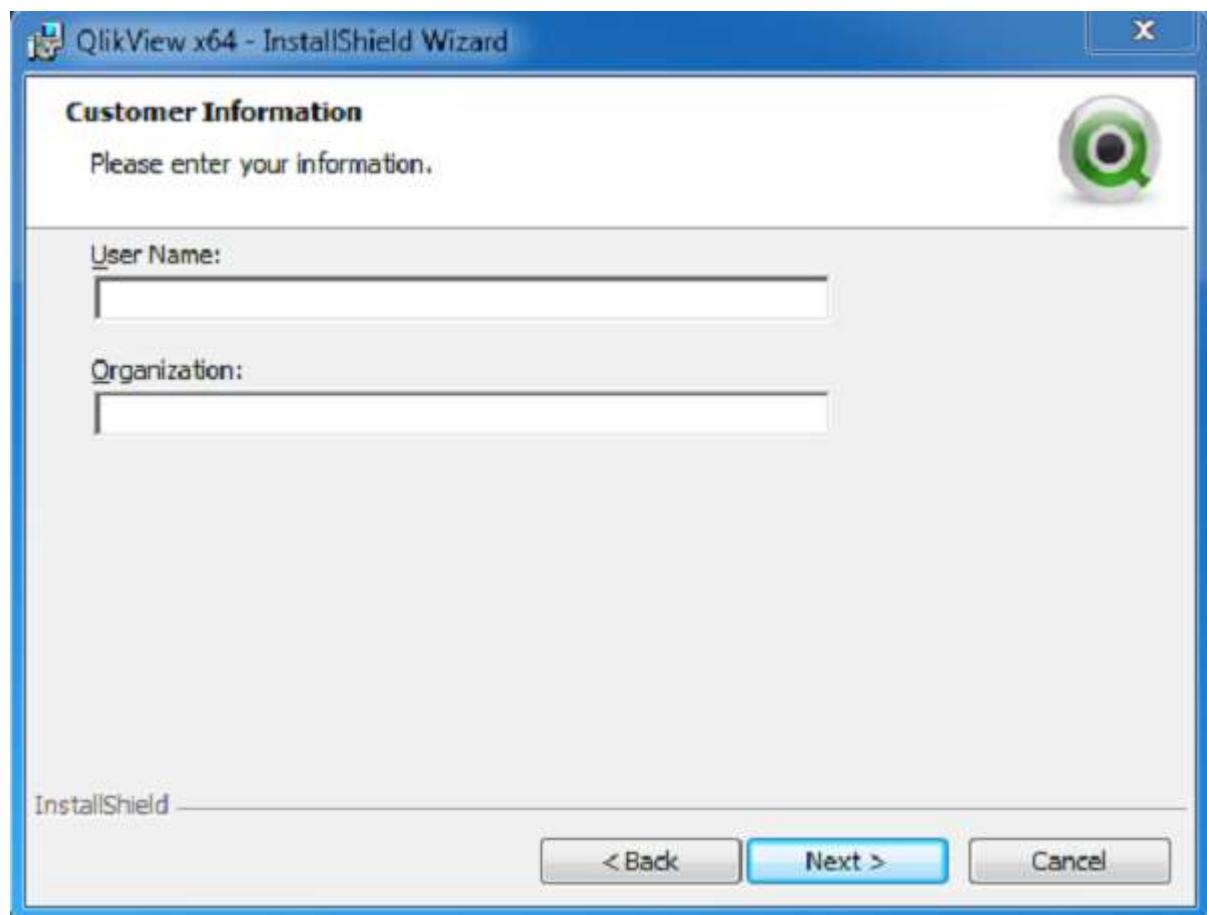
### Accept the License Agreement

Read the license agreement and if you agree, choose the "I accept the terms in the license agreement" option. Then click "Next".



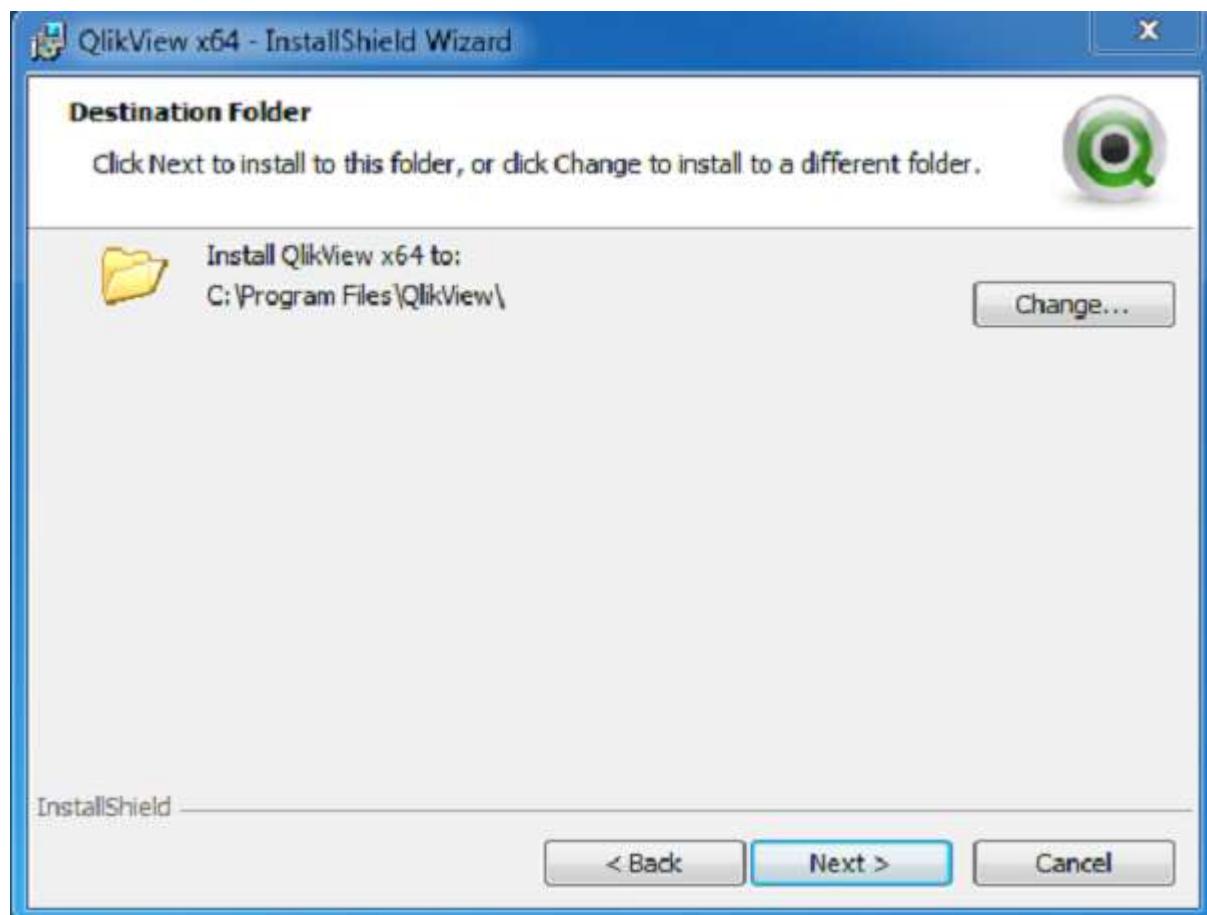
## Provide your Details

Provide your name and organization details. Then Click "Next".



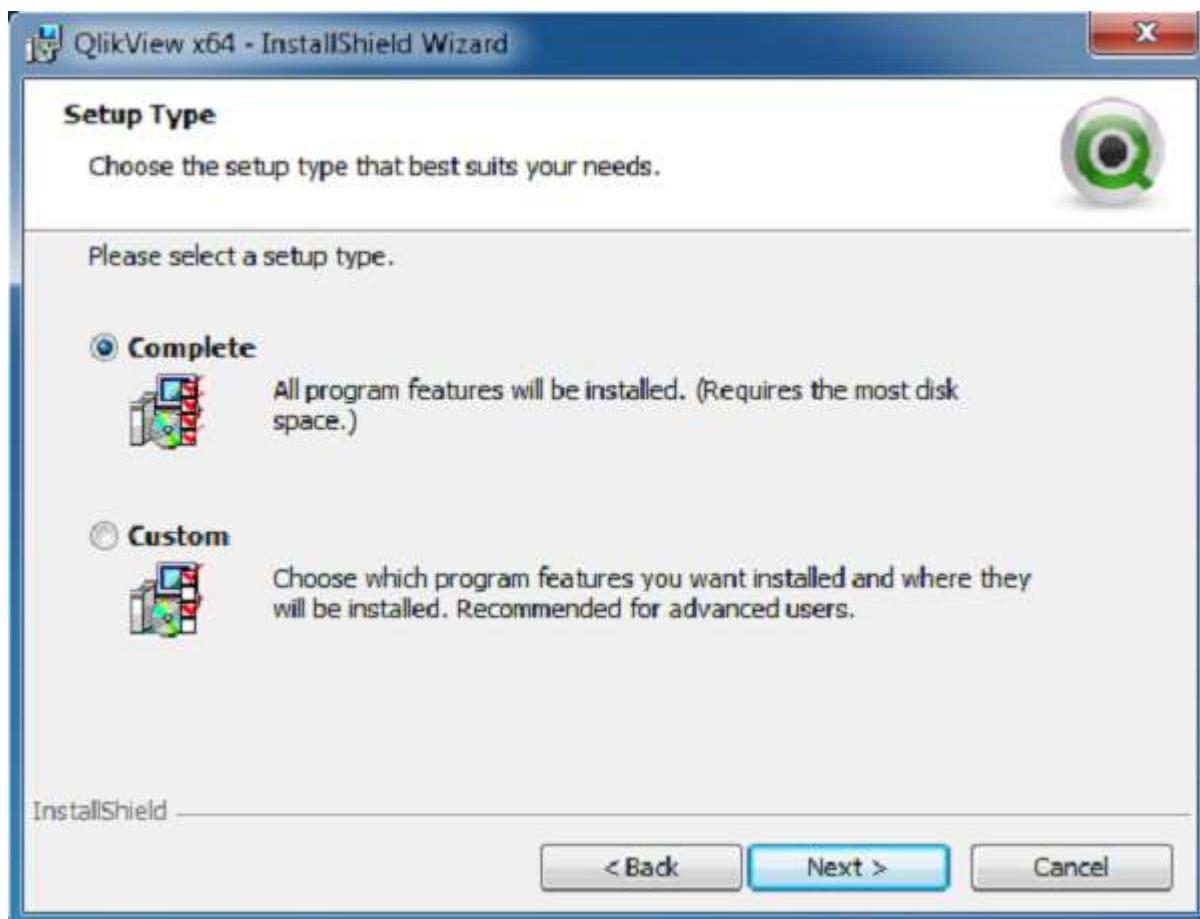
## Choose the Destination Folder

You may accept the default destination for installation or alter it. Then click "Next".



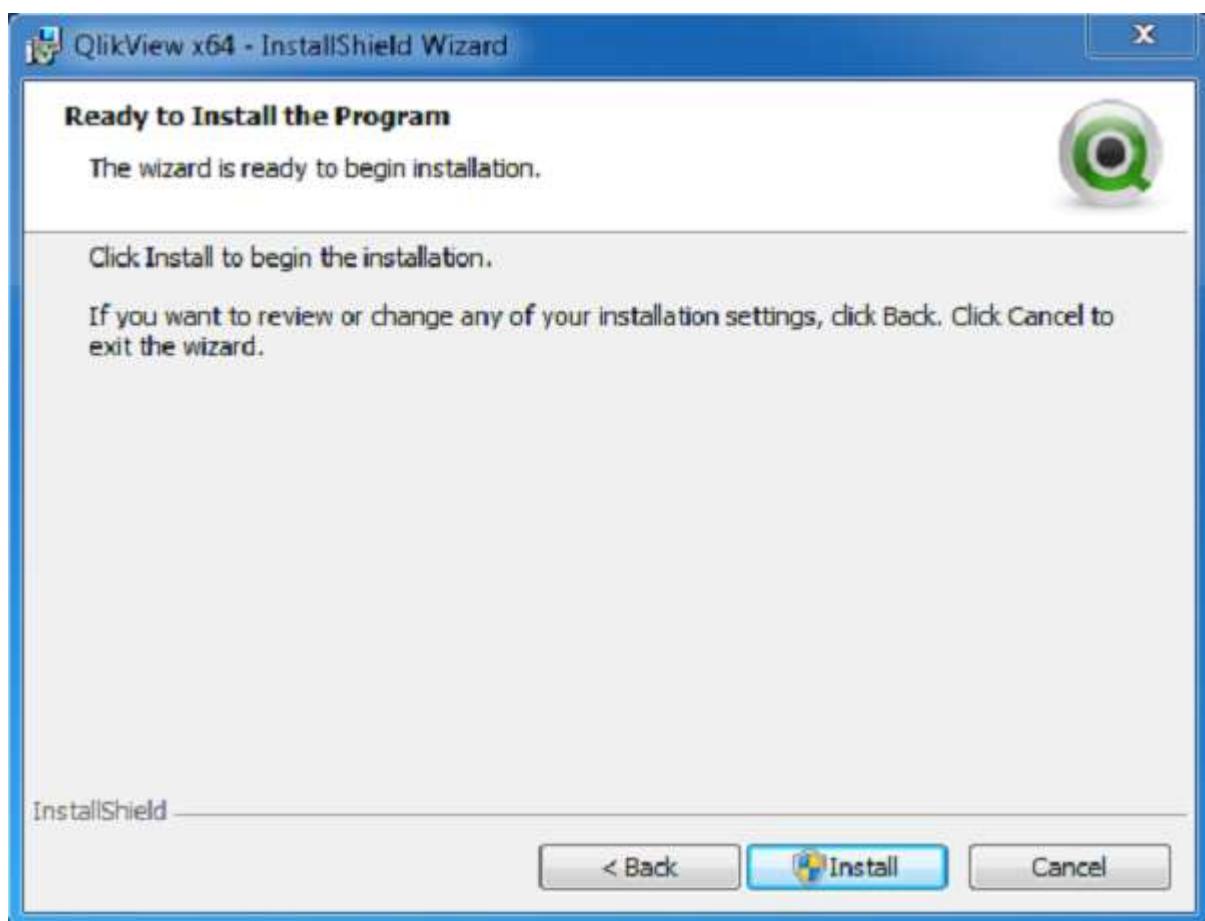
## Choose the Setup Type

Choose the setup type as "Complete". Then click "Next".



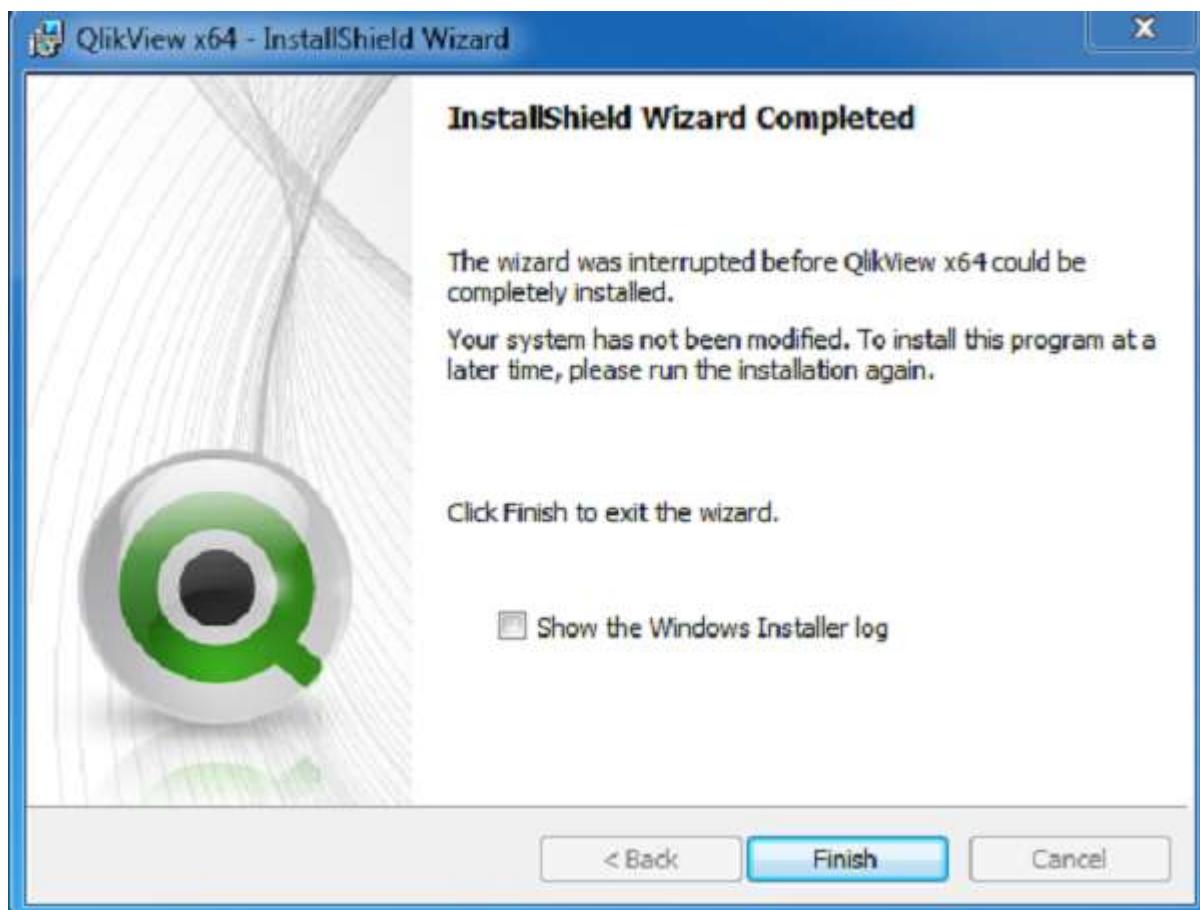
## Begin Installation

In this screen, you finally decide to really start the installation. You can still go back and change some options if needed. Assuming you are fine with everything so far, click "Install".



## Installation Complete

The installation completion screen appears after successful installation. Click "Finish".



## Verify the Installation

You can verify the installation by going to the Windows Start menu and clicking on the QlikView icon. The screen appears as shown below.



You are now ready to learn QlikView.

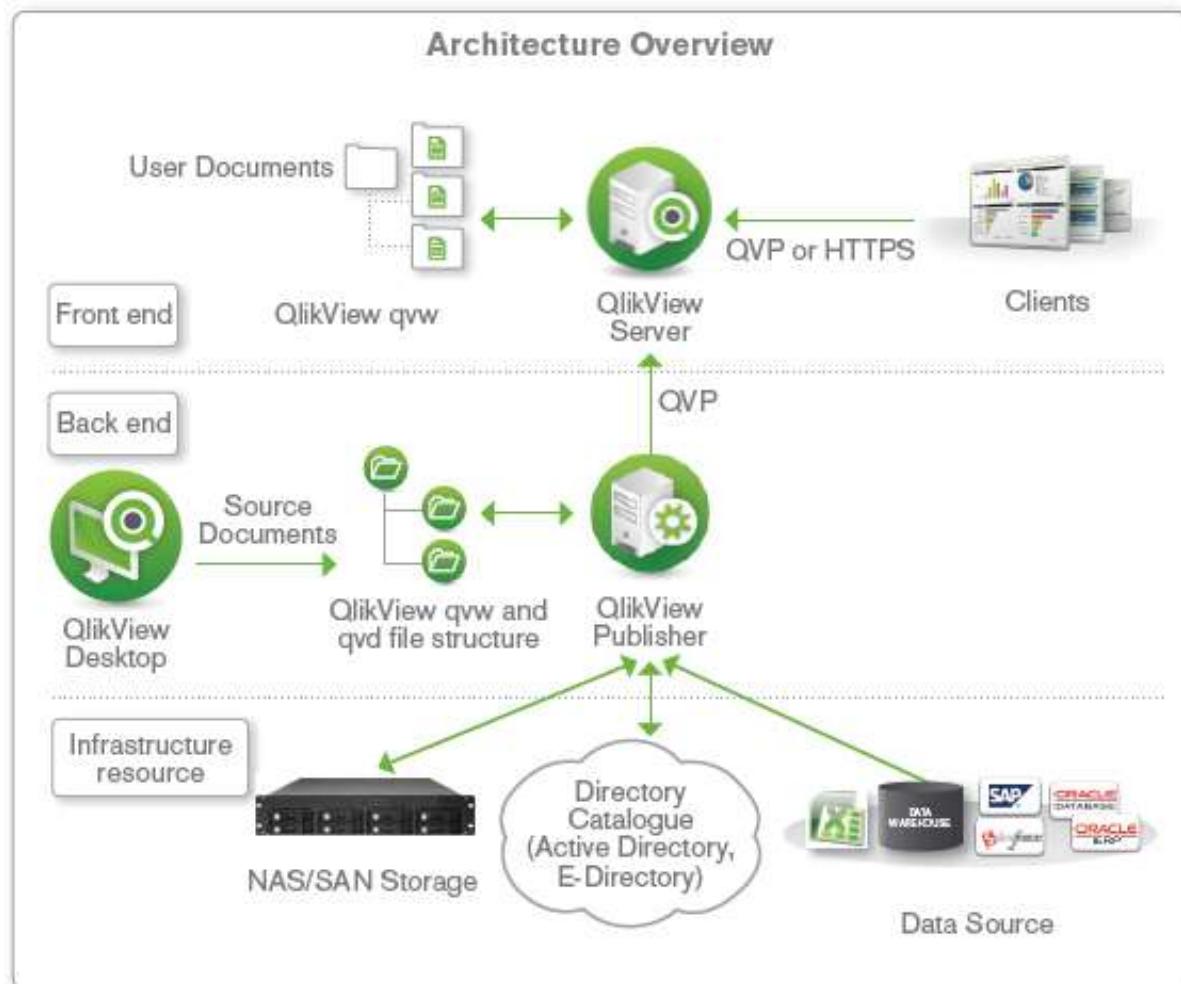
### 3. QlikView – Architecture

As a leading Business Discovery Platform, QlikView is built with a very different approach to data discovery than other traditional platforms. QlikView does not first build a query and then fetch the result based on the query. Rather, it forms associations between different data objects as soon as it is loaded and prompts the user to explore the data in any way. There is no predefined data drill down paths. The data drill down paths can happen in any direction as long as the data is available and associated.

Of course, a user can also play a role in creating the associations between data elements using data modeling approach available in QlikView.

#### Architectural Overview

QlikView's architecture consists of a front end to visualize the processed data and a back end to provide the security and publication mechanism for QlikView user documents. The diagram given below depicts the internal working of QlikView. The architecture is discussed in detail below the picture.



## Front End

The Front end in QlikView is a browser-based access point for viewing the QlikView documents. It contains the **QlikView Server**, which is mainly used by the Business users to access the already created BI reports through an internet or intranet URL.

Business users explore and interact with data using this front end and derive conclusions about the data. They also collaborate with other users on a given set of reports by sharing insights and exploring data together, in real time or off-line. These user documents are in the format .qvw, which can also be stored in the windows OS as a standalone document.

The QlikView server in the front end manages the client server communication between the user and QlikView backend system.

## Back End

The QlikView backend consists of **QlikView desktop** and **QlikView publisher**.

The **QlikView desktop** is a wizard-driven Windows environment, which has the features to load and transform data from its source. Its drag and drop feature is used to create the GUI layout of the reports that becomes visible in the frontend. The file types, which are created by QlikView desktop are stored with an extension of **.qvw**. These are the files that are passed on to the QlikView server in the front end, which serves the users with these files. Also **.qvw** files can be modified to store the data-inly files, which are known as **.qvd** files. They are binary files, which contain only the data and not the GUI components.

The **QlikView publisher** is used as distribution service to distribute the **.qvw** documents among various QlikView servers and users. It handles the authorization and access privileges. It also does the direct loading of data from data sources by using the connection strings defined in the **.qvw** files.

## 4. QlikView – Navigation

In this chapter, we will get acquainted with the screens available to a user for doing various operations. We will learn the basic navigation and know the different functions performed by the icons available in QlikView.

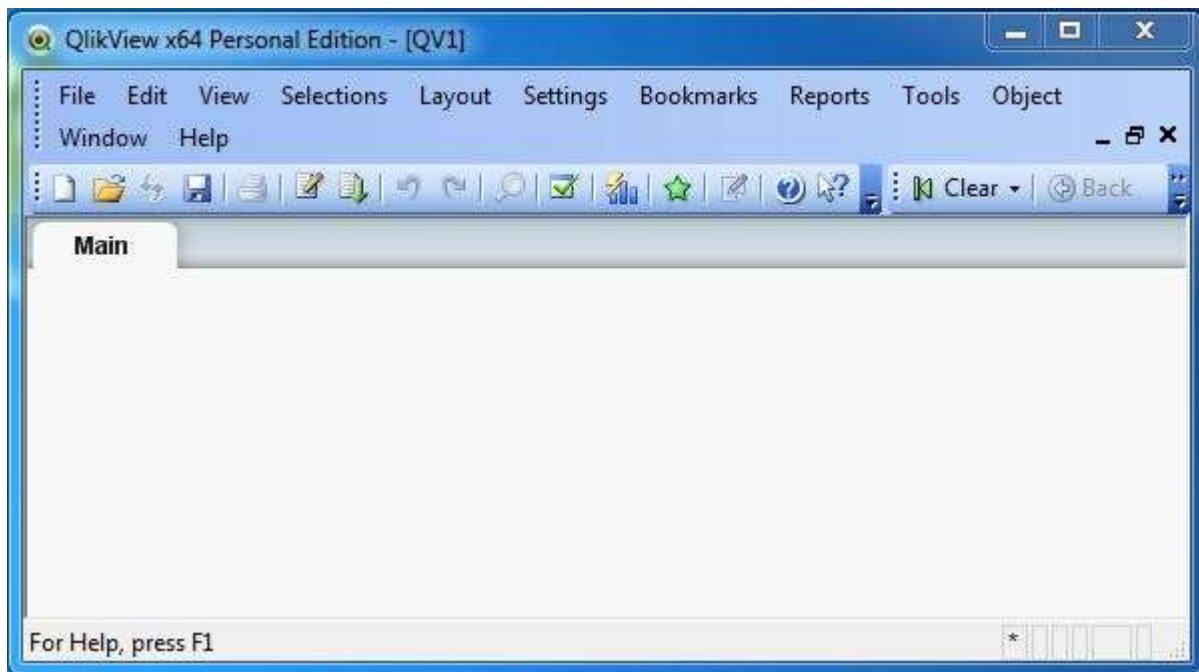
### Getting Started Screen

This screen is a gentle introduction to navigate around QlikView. This screen comes up when you start QlikView and keep the **Show start page when launching QlikView** option checked. If you scroll down the examples section in the left, you can click any of the examples like - **Movies Database**, **Data Visualization** etc. to invoke it and see how QlikView works. Feel free to click around! On moving to the right, you notice the **Recent** and **Favourites** link, which show all the recently visited QlikView documents and the documents you want to visit frequently.



## Menu Commands

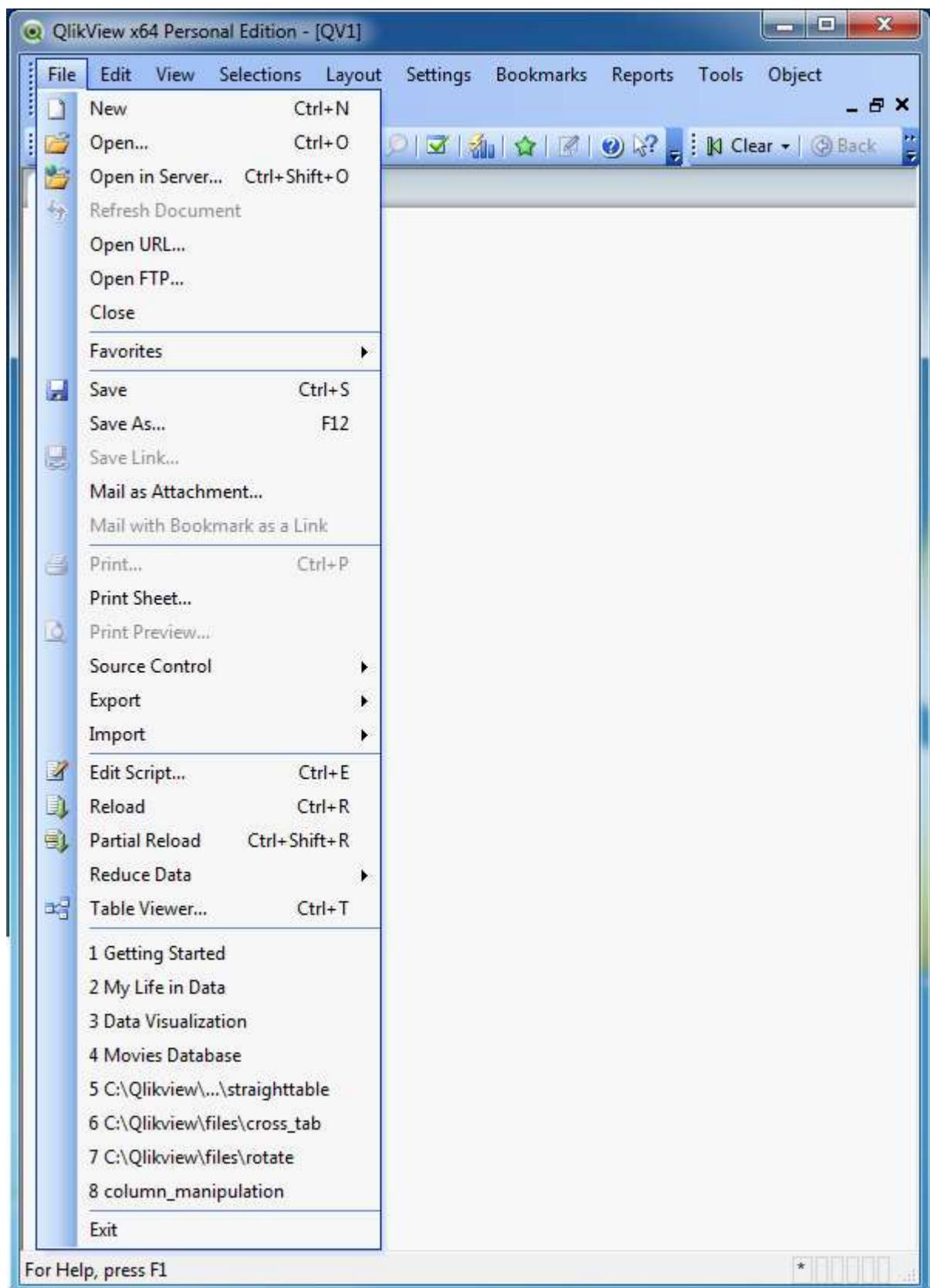
On closing the 'Getting Started' window, we get the main interface with all the available Menu commands. They represent the entire set of features available in QlikView. Given below is an overview of each section of the Menu Commands.



### File Menu

This menu is used to create a new QlikView file and open the existing files from both local system and QlikView server. The important features in this menu are:

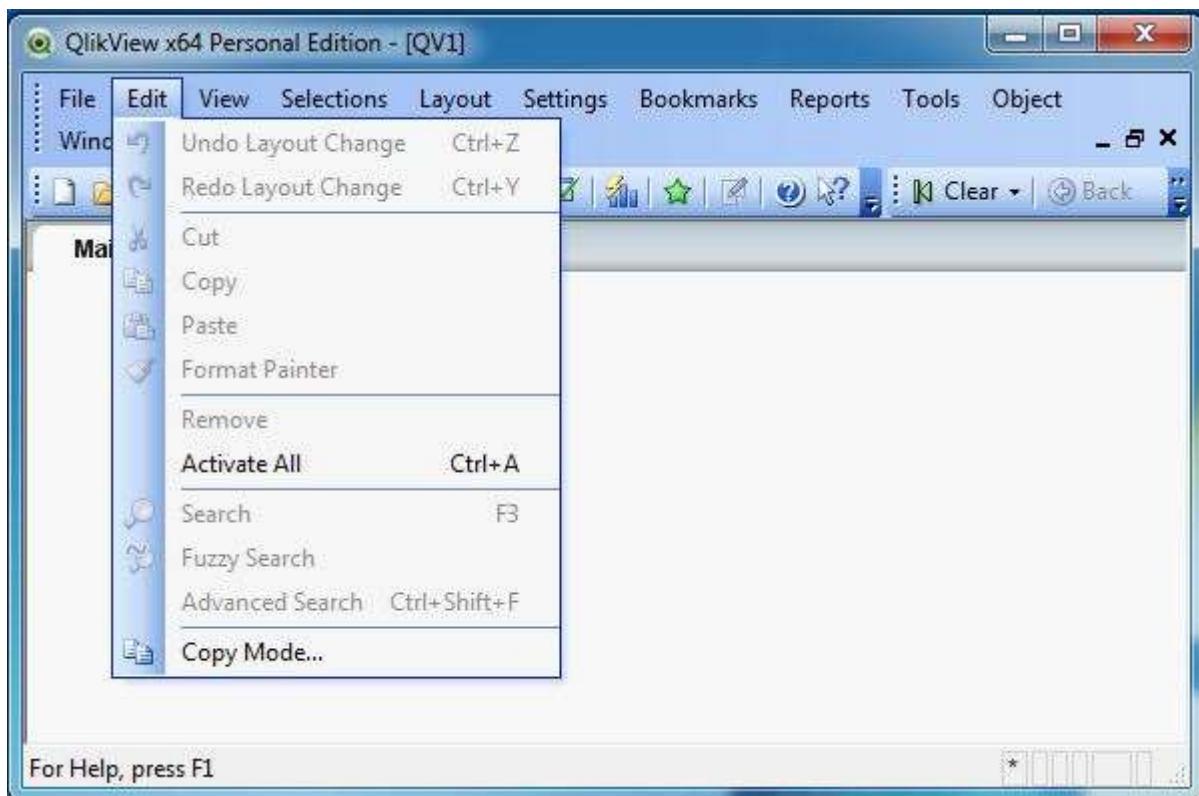
- **Mail as attachment** to email the currently open document as an attachment.
- **Reduce Data** to view only the report layout and database structure without any data.
- **Table viewer** option is used to see the structure of the tables, fields and their association in a graphical view.



## Edit Menu

This menu is used to carry out the editing options like copy, paste, cut and using format painter. The important features in this menu are:

- **Active All** option activates all the sheet objects present in the opened sheet.
- **Remove** to remove a sheet from the active window.
- **Advanced search** option is used to do a search with advanced search expressions using multi box.



## View Menu

This menu is used to view the standard toolbars and zoom in/ zoom out features. It also displays all the active sheets as a cascade menu. The important features in this menu are:

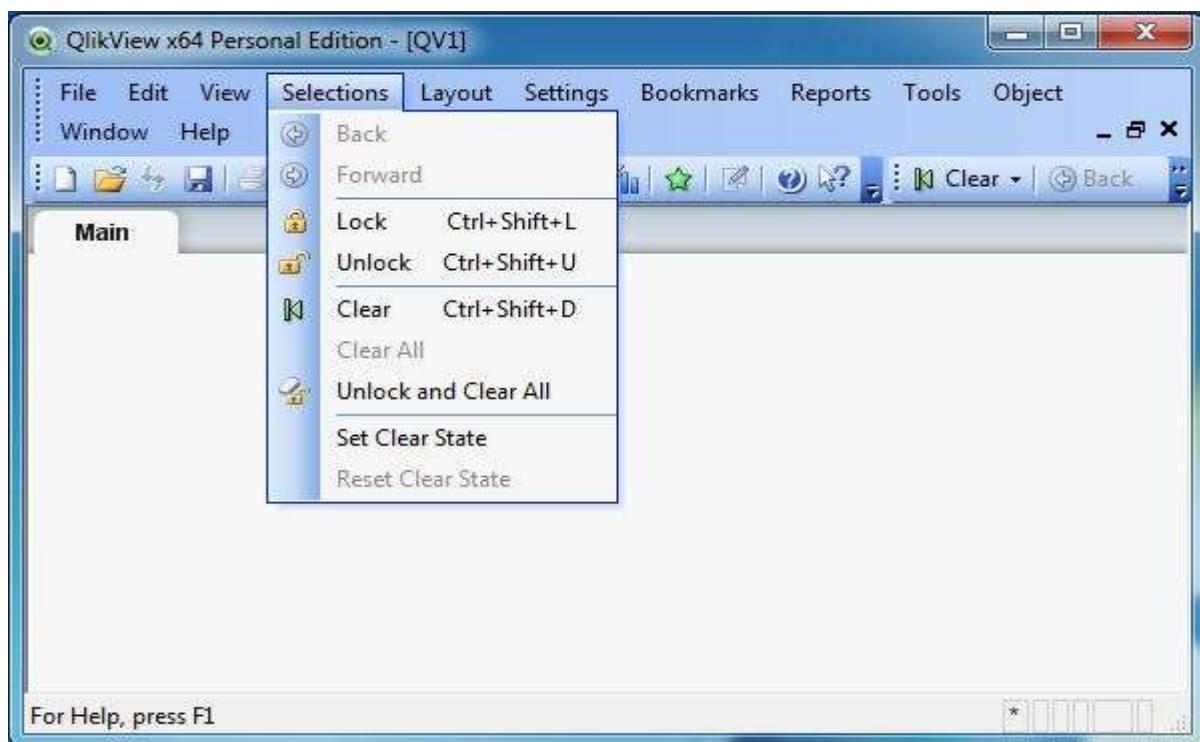
- **Turn on/off WebView mode** toggles the WebView mode and local view mode.
- **Current Selections** displays the field name and file values of the selected sheet objects.
- **Design Grid** is used to toggle the sheet object placeholders for active object(s) and snap-to-grid for sizing and moving objects in the layout.



## Selection Menu

This menu is used to select and clear the selection of values in the sheet objects. It also provides the feature of going back and forward into different logical statements of the sheet, you are working on. The important features in this menu are:

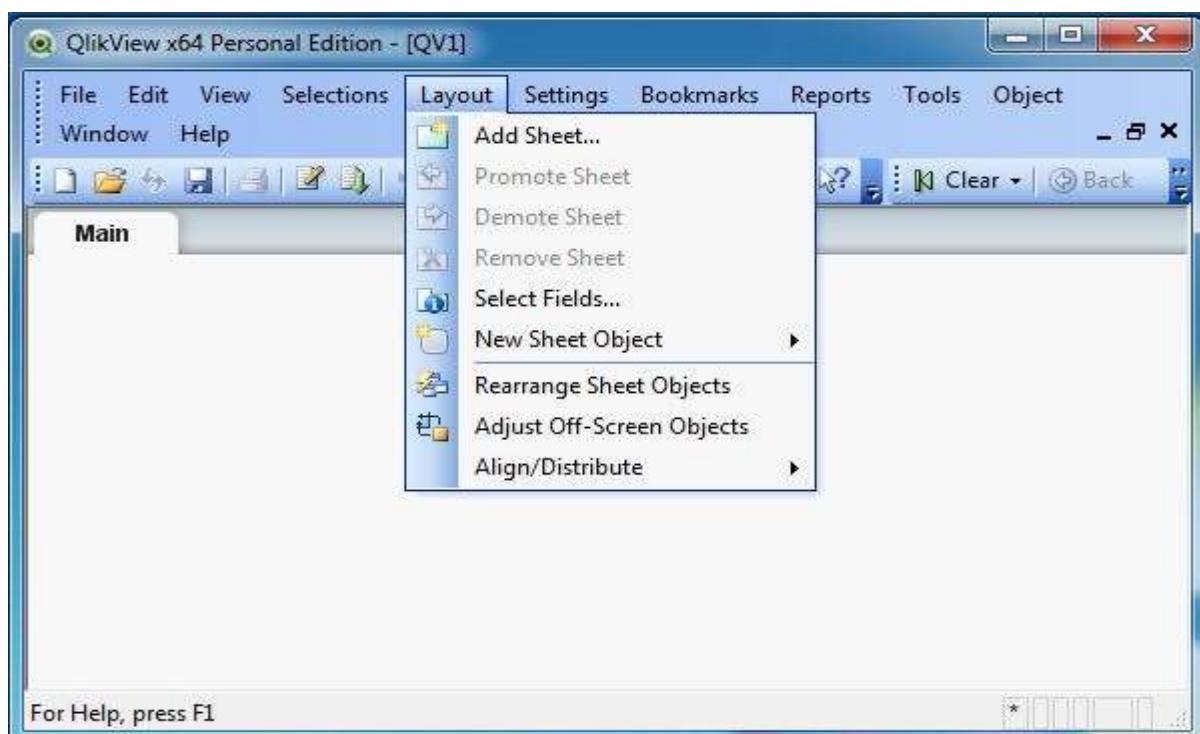
- **Lock** locks all the values in current selection.
- **Unlock** unlocks all the locked values in the current selection.



## Layout Menu

Layout Menu is used to add tabbed sheets, select different sheets and rearrange sheet objects. The important features in this menu are:

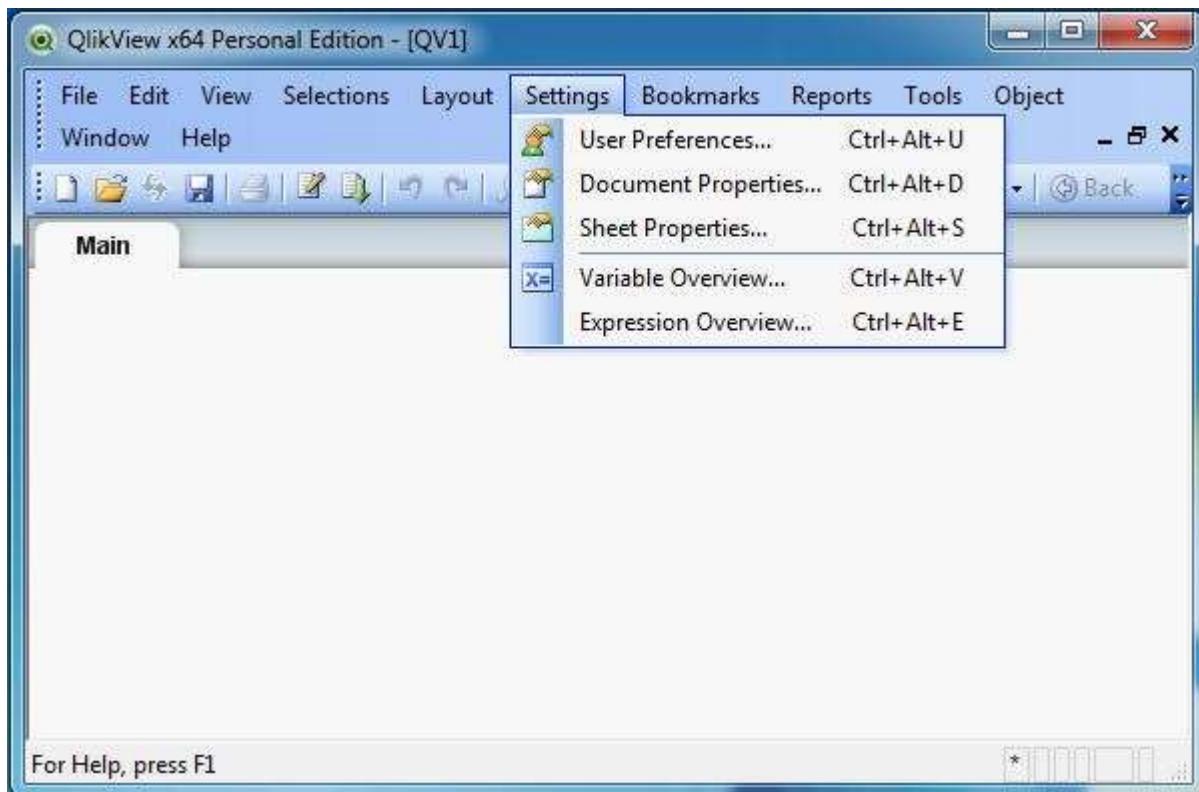
- **Promote sheet** moves the current sheet or tab one step forward.
- **Demote sheet** moves the current sheet or tab one step backward.
- **Delete sheet** deletes the active sheet and everything in it.



## Settings Menu

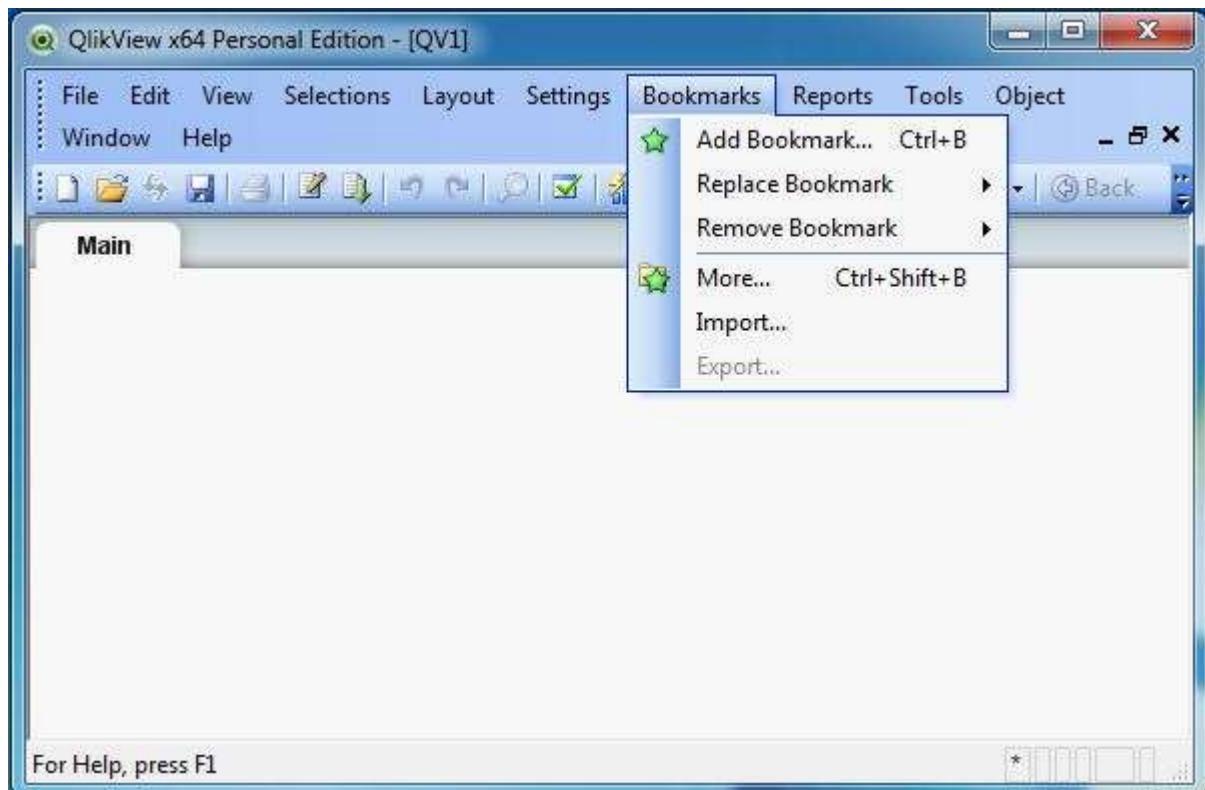
Settings menu is used to set the user preferences, document properties, and sheet properties. The important features in this menu are:

- **Variable overview** shows all the non-hidden variables and their values in a single list.
- **Expression Overview** shows expressions from the document, sheet and sheet objects as a single list.



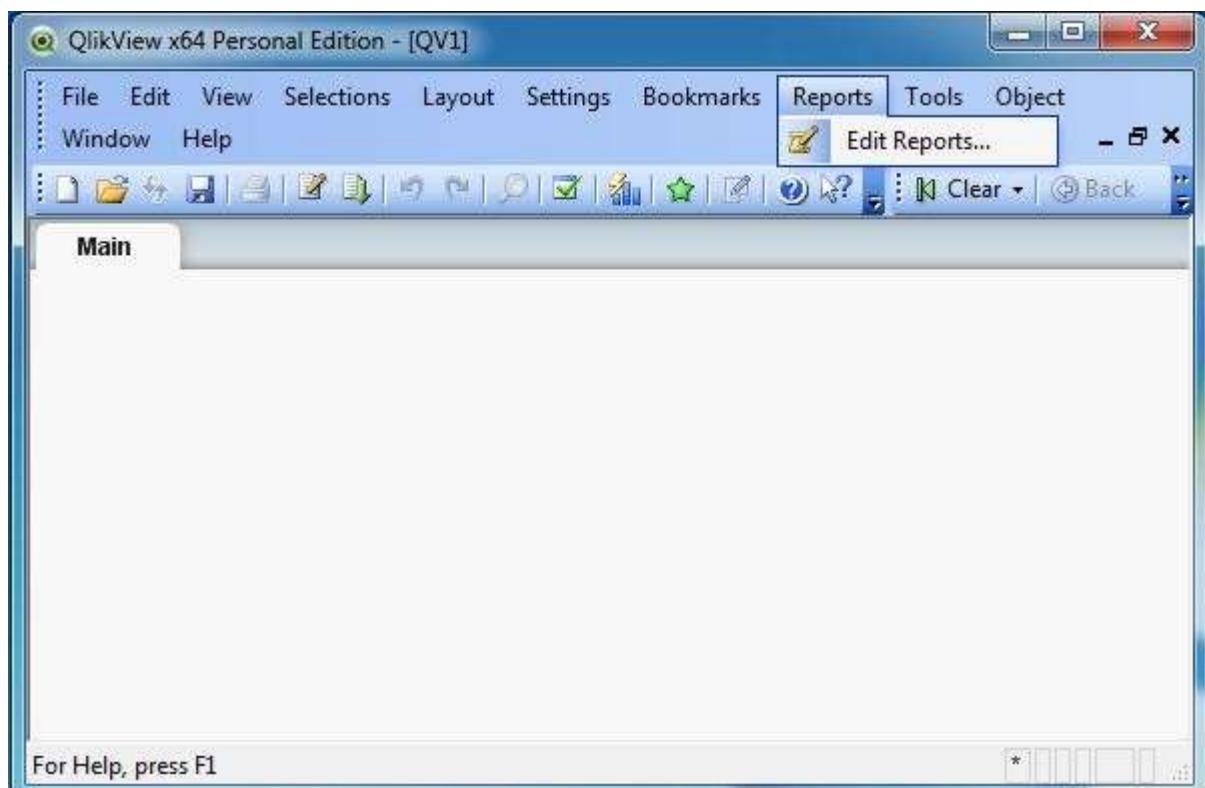
## Bookmark Menu

This menu is used to create bookmarks to different documents for faster retrieval.



## Reports Menu

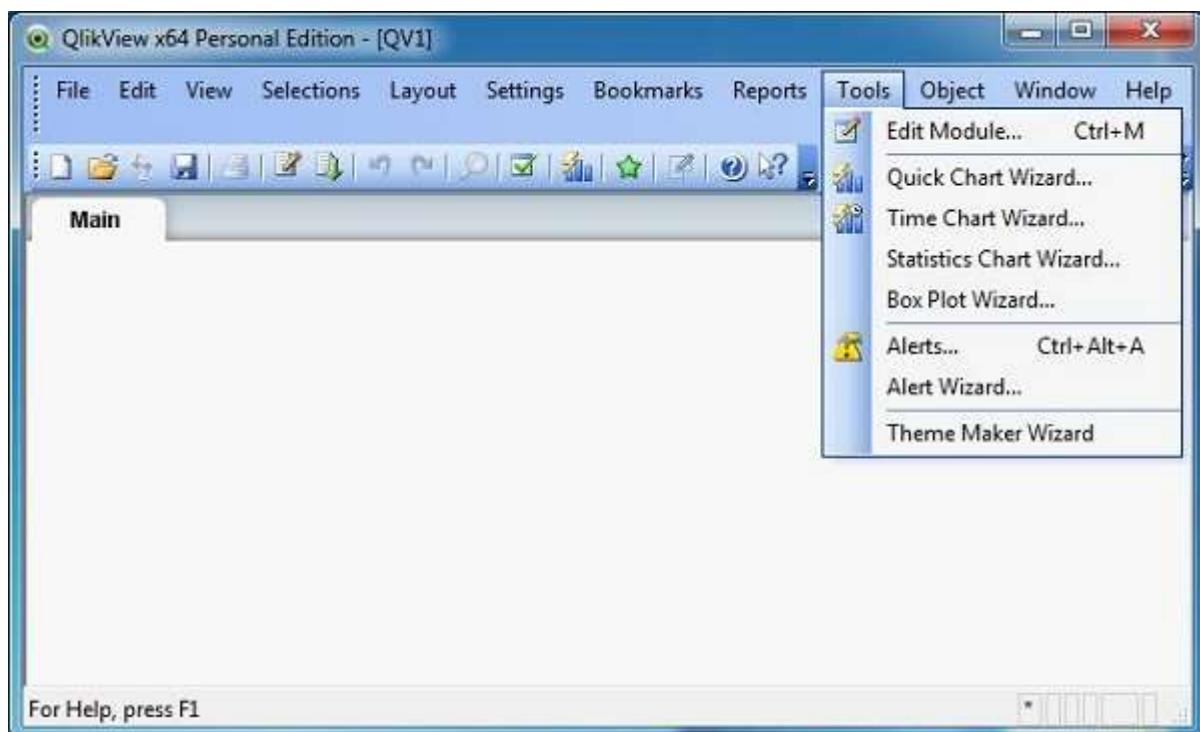
Reports menu is used to create new reports and edit the existing reports. You can edit the layout, add pages to the report, and also delete reports.



## Tools Menu

Tools menu is a very prominent menu, frequently used for creating charts and opening the QlikView management console. The important features in this menu are:

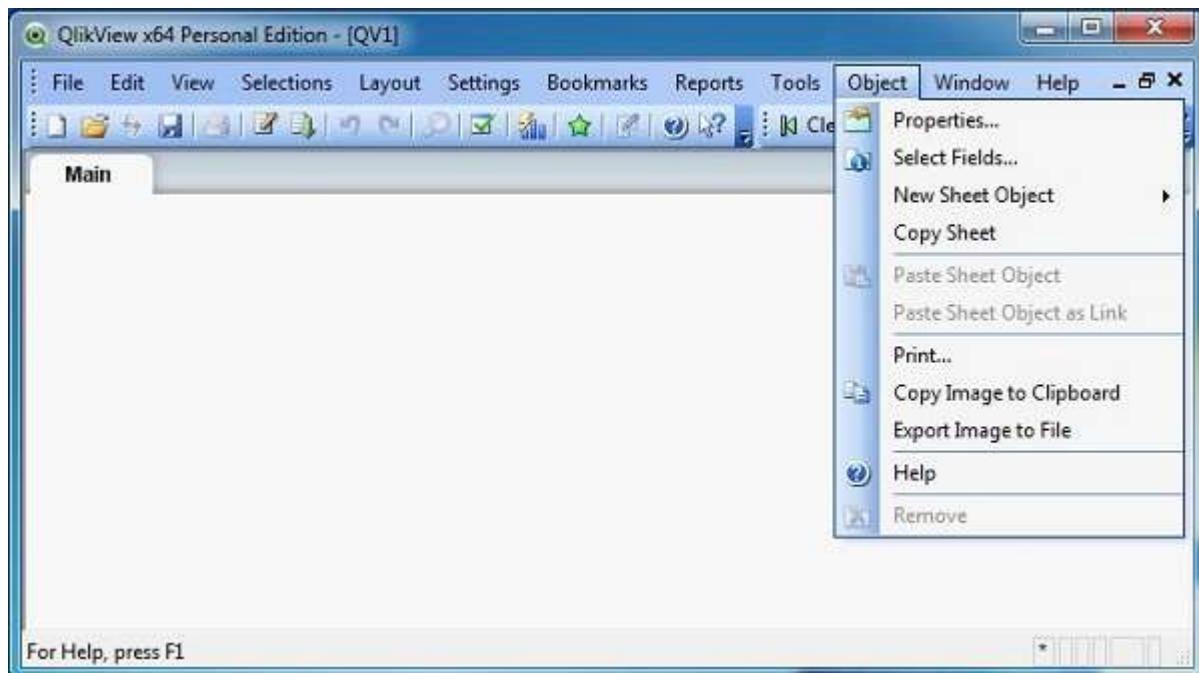
- **Quick Chart Wizard** creates simple charts without using the great number of different settings and options available.
- **Time Chart Wizard** creates time series charts.
- **Statistics Chart Wizard** is used to apply common statistical tests on data.



## Object Menu

This menu is used to create new sheet objects and modify the existing ones. The sheet properties option opens the page to set the parameters defining the sheet. The important features in this menu are:

- **Copy Sheet** - Creates a copy of the sheet along with all the sheet objects.
- **Copy Image to Clipboard** - Copies a bitmap picture of the sheet area to Clipboard.
- **Remove** - Completely removes the sheet along with the sheet objects.



The **Window** and **Help** menus are used to organize the different windows of QlikView application and provide help documentation.

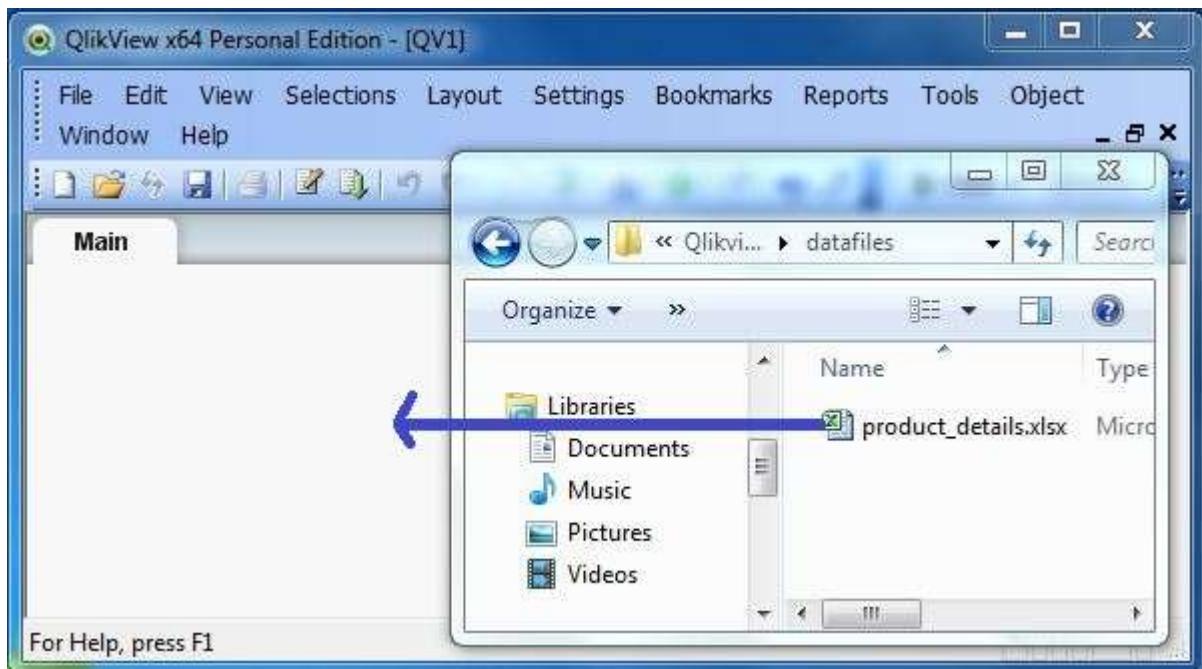
# QlikView Data Loading

## 5. QlikView – Excel Files

QlikView accepts Excel spreadsheet for data analysis by simple drag and drop action. You need to open the QlikView main window and drag and drop the excel file into the interface. It will automatically create the sheet showing the excel data.

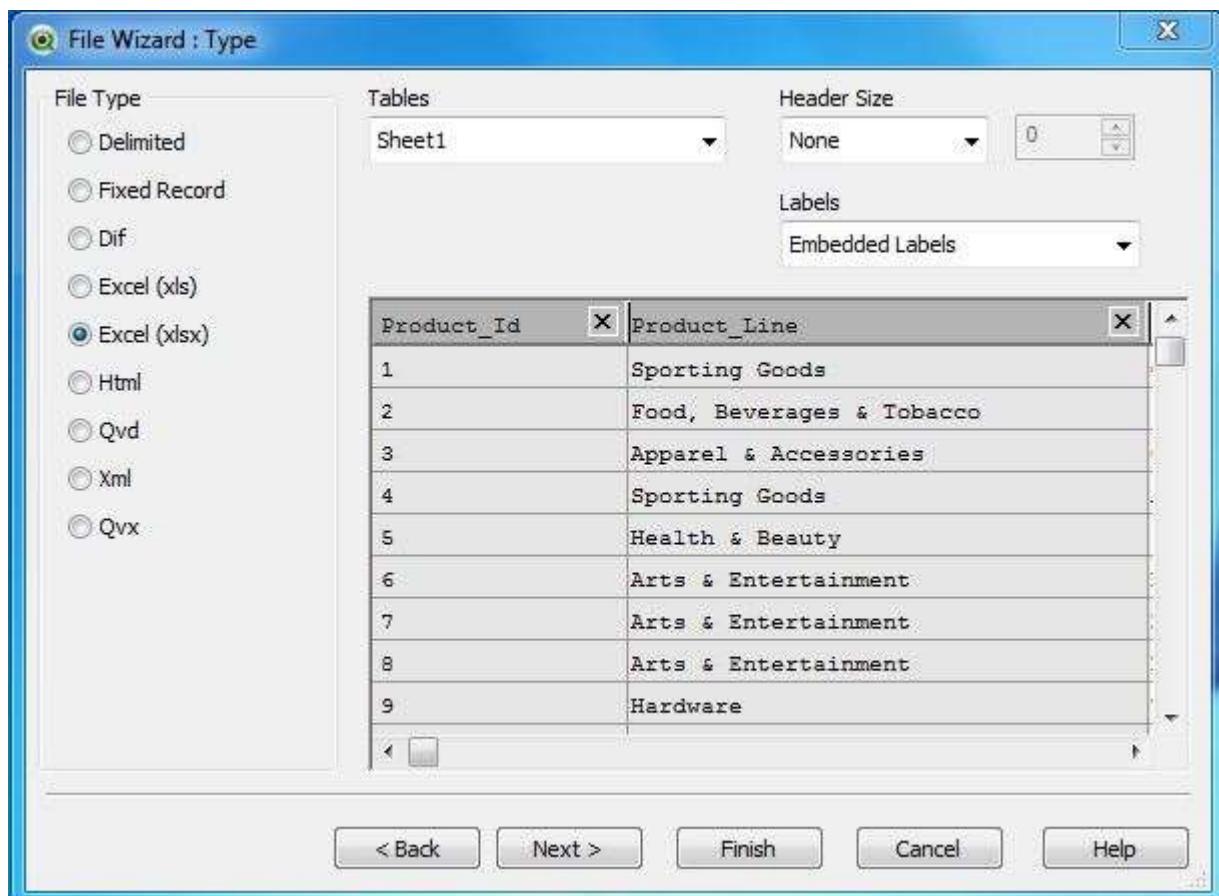
### Select the Excel file

Keep the main window of QlikView open and browse for the excel file you want to use.



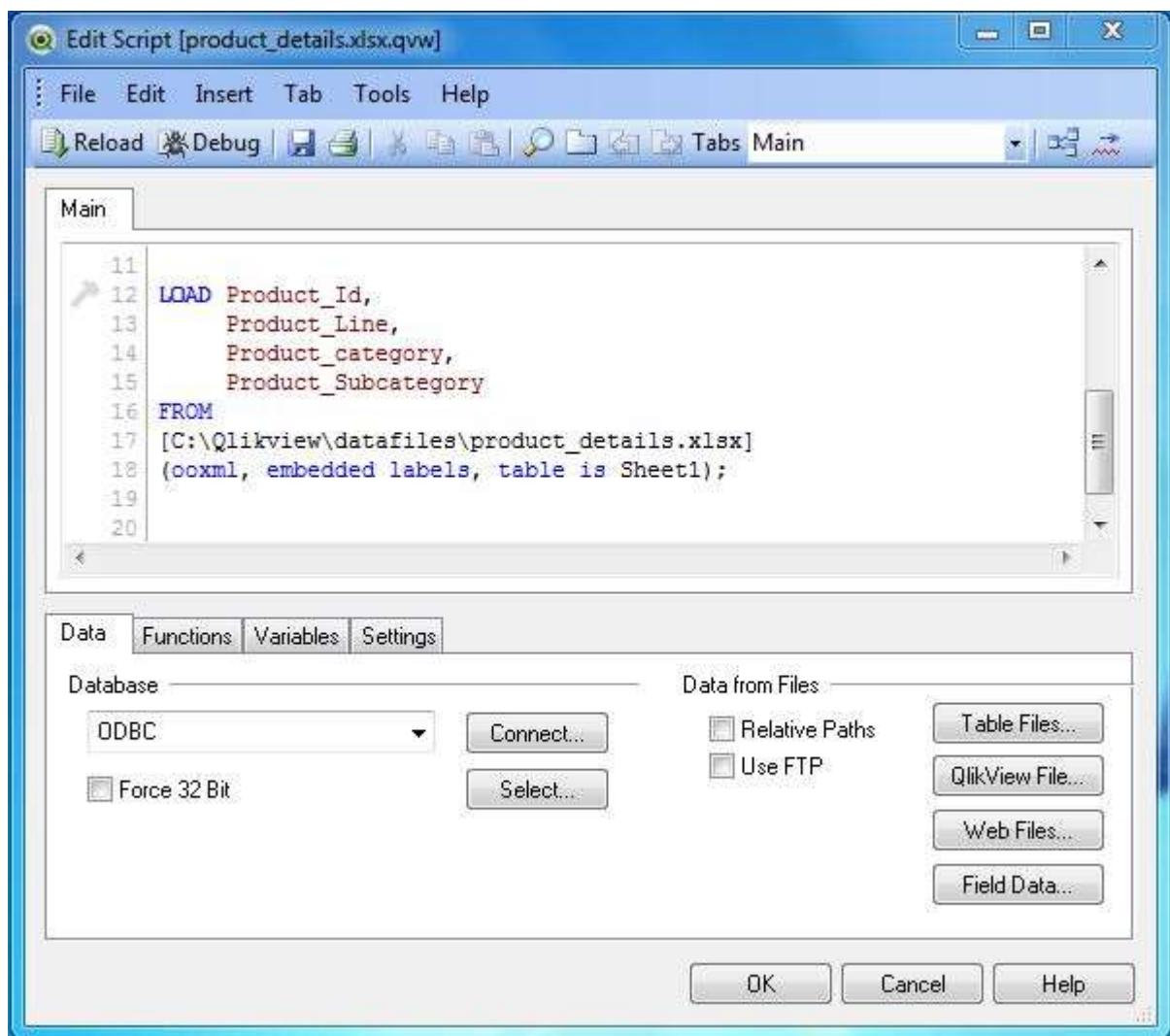
## Select a Data Source

On dropping the excel file into the main window, the File wizard appears. The File Type is already chosen as Excel. Under **Labels**, choose **Embedded Labels**. Click "Next step" to proceed.



## Load Script

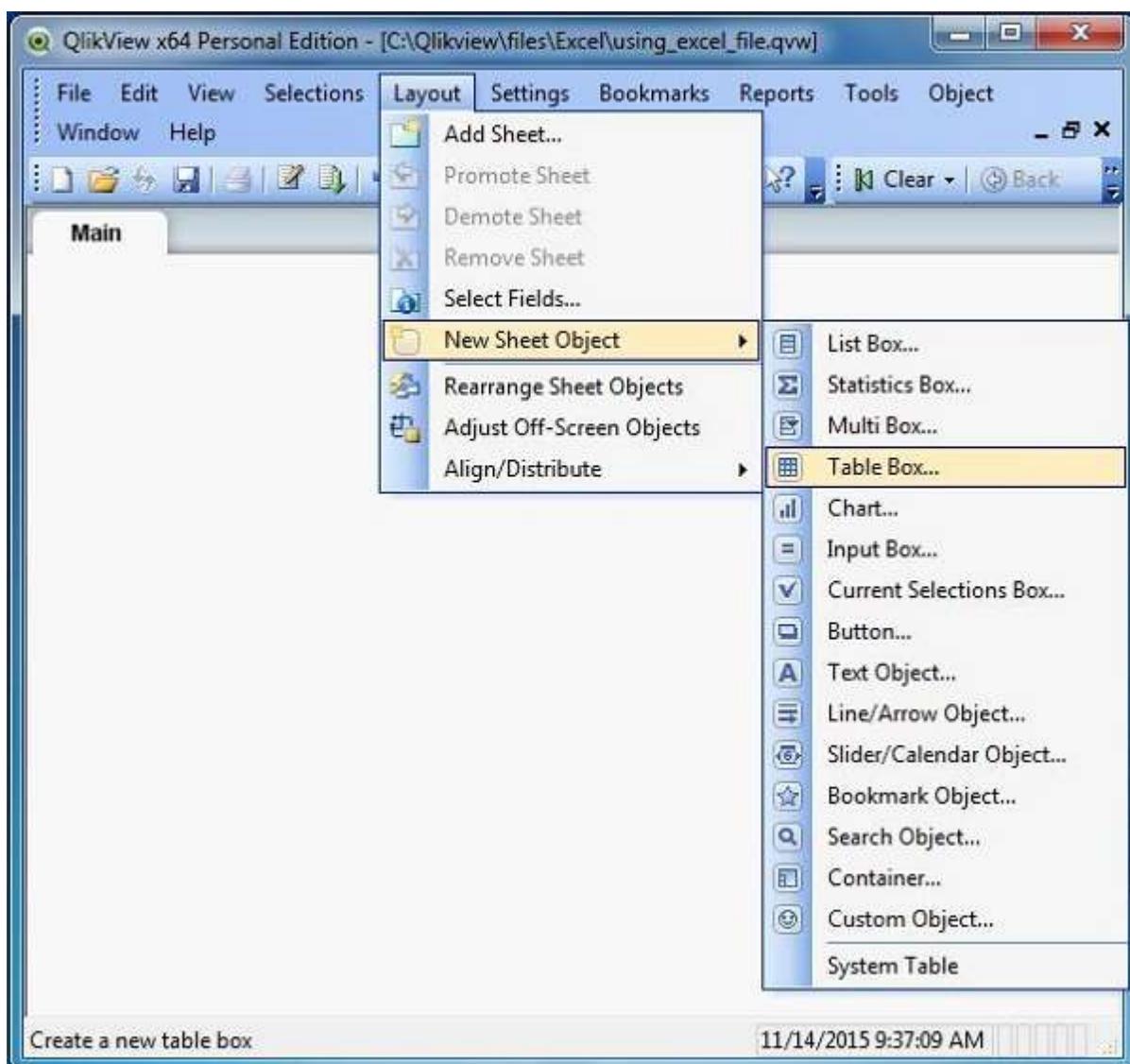
The Load script appears which shows the command that loads the data into the QlikView document. This command can be edited.



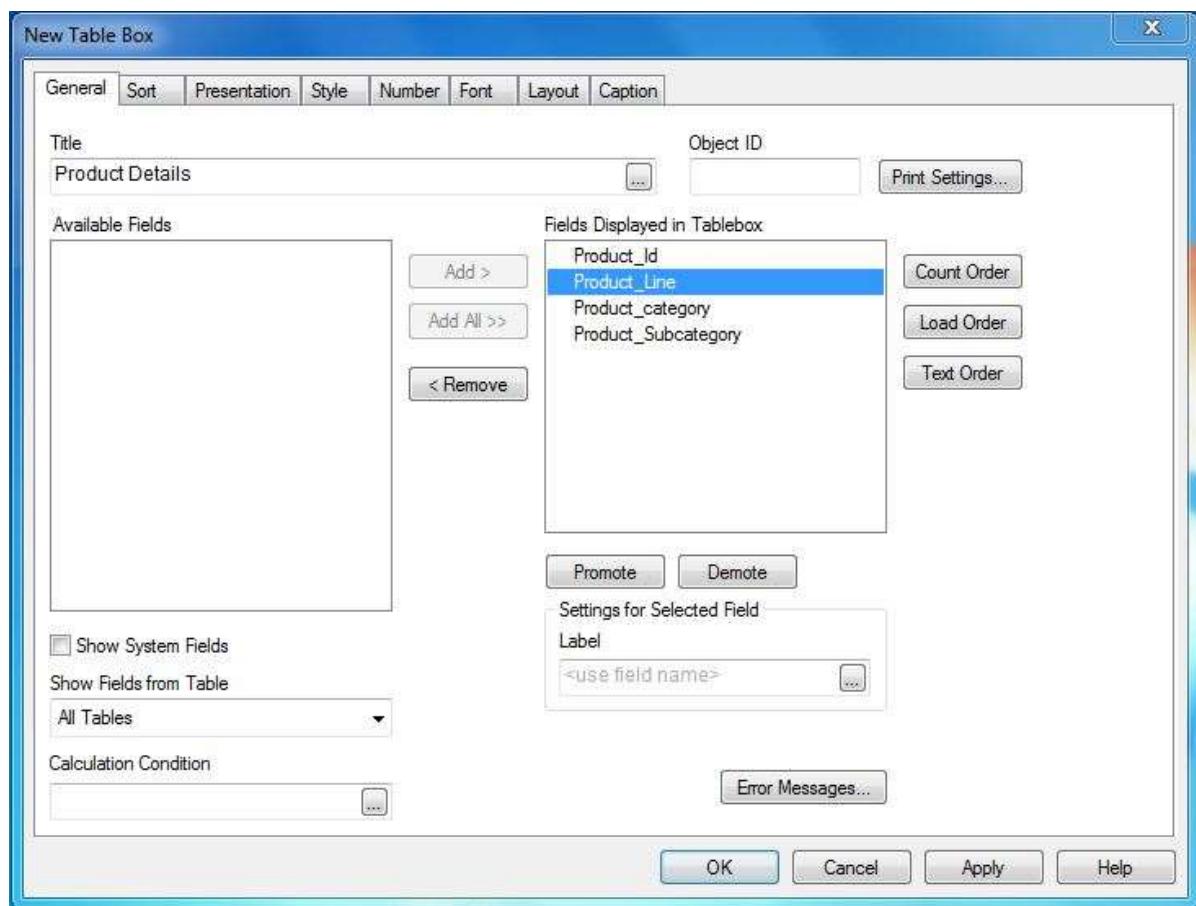
Now, the Excel wizard prompts to save the file in the form of \*.qvw file extension. It asks to select a location where you need to save the file. Click "Next step" to proceed. Now it is time to see the data that is loaded from the Excel file. We use a **Table Box** sheet object to display this data.

## Create Table Box

The **Table Box** is a sheet object to display the available data as a table. It is invoked from the menu **Layout -> New Sheet Object -> Table Box**.



On clicking Next, we get the option to choose the fields from the Table Box. You can use the **Promote or Demote** buttons to rearrange the fields.



## Table Box Data

On completing the above step, the Table Box Sheet Object appears which shows the data that is read from the Excel file.

QlikView x64 Personal Edition - [C:\Qlikview\files\Excel\using\_excel\_file.qvw]

File Edit View Selections Layout Settings Bookmarks Reports Tools Object Window Help

Main

**Product Details**

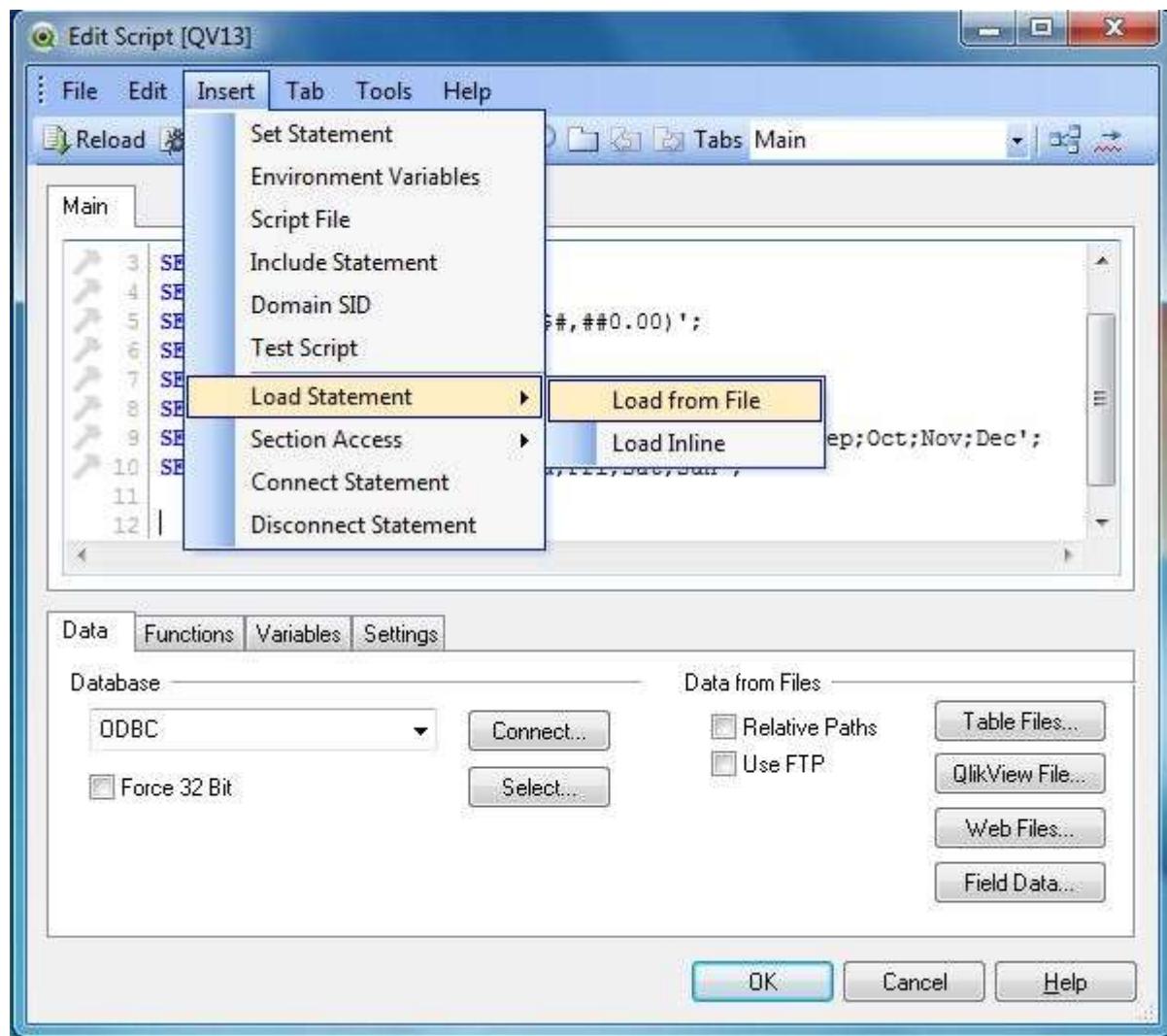
Product_Id	Product_Line	Product_category	Product_Subcategory
1	Sporting Goods	Outdoor Recreation	Winter Sports & Activities
2	Food, Beverages & Tobacco	Food Items	Fruits & Vegetables
3	Apparel & Accessories	Clothing	Uniforms
4	Sporting Goods	Athletics	Rugby
5	Health & Beauty	Personal Care	-
6	Arts & Entertainment	Hobbies & Creative Arts	Musical Instruments
7	Arts & Entertainment	Hobbies & Creative Arts	Orchestra Accessories
8	Arts & Entertainment	Hobbies & Creative Arts	Crafting Materials
9	Hardware	Tool Accessories	Power Tool Batteries
10	Home & Garden	Bathroom Accessories	Bath Caddies
11	Food, Beverages & Tobacco	Food Items	Frozen Vegetables
12	Home & Garden	Lawn & Garden	Power Equipment

For Help, press F1 11/14/2015 9:37:09 AM 12 X 4

## 6. QlikView – Delimited File

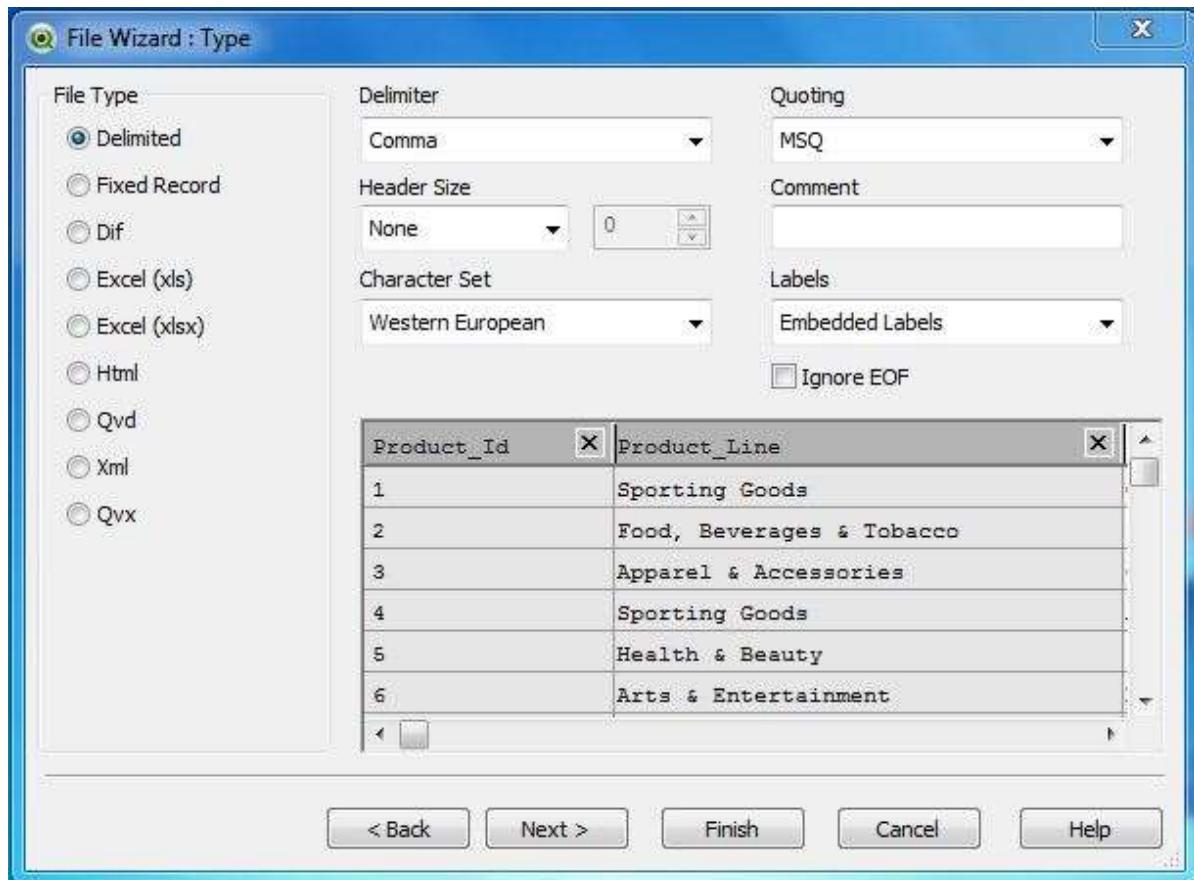
QlikView can use the data in plain text file where the fields are separated by characters like comma, tab, semicolon etc. Here, we will take CSV as an example. A file in which each column of data is separated by a comma is known as a CSV file. It is a very widely used file format to store plain text-data organized as columns and rows.

QlikView loads CSV files using the **Data from files** options available in the script editor under the File Menu. Alternatively, you can also open a new QlikView document and press **control+E** to get the script editor window as shown below. Choose the file Product\_details.csv from the appropriate path.



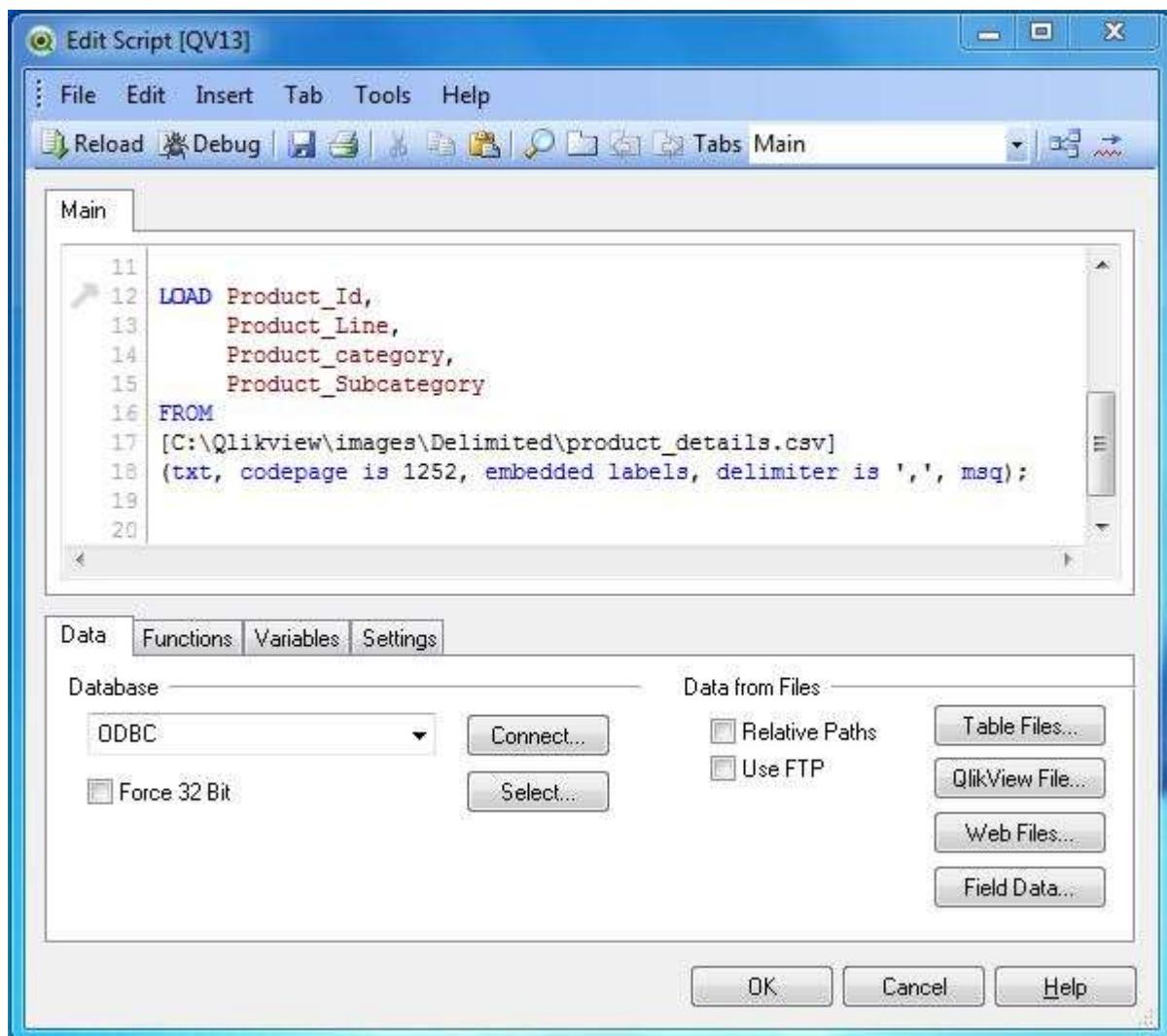
## Select the File Headers

On opening the selected CSV file, a window as shown below comes up. Under **Labels**, choose **Embedded Labels**, as our file has a header row as its first row. Click **Finish**.



## Load Script

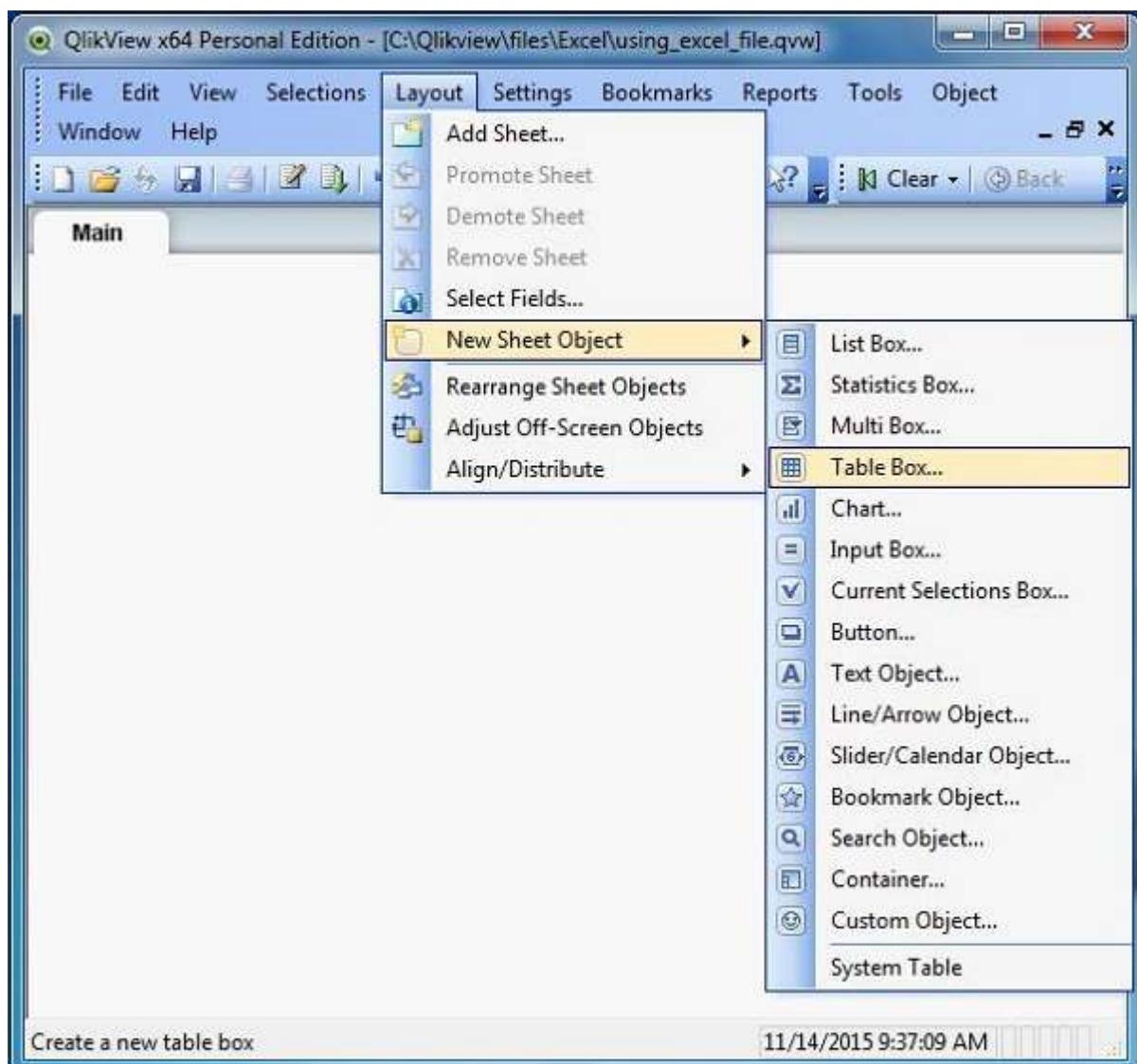
The loading of the file into QlikView is done through the load script, which can be seen in the screen shot below. Hence, when we use any delimited file, we can tweak the below script as per the file format.



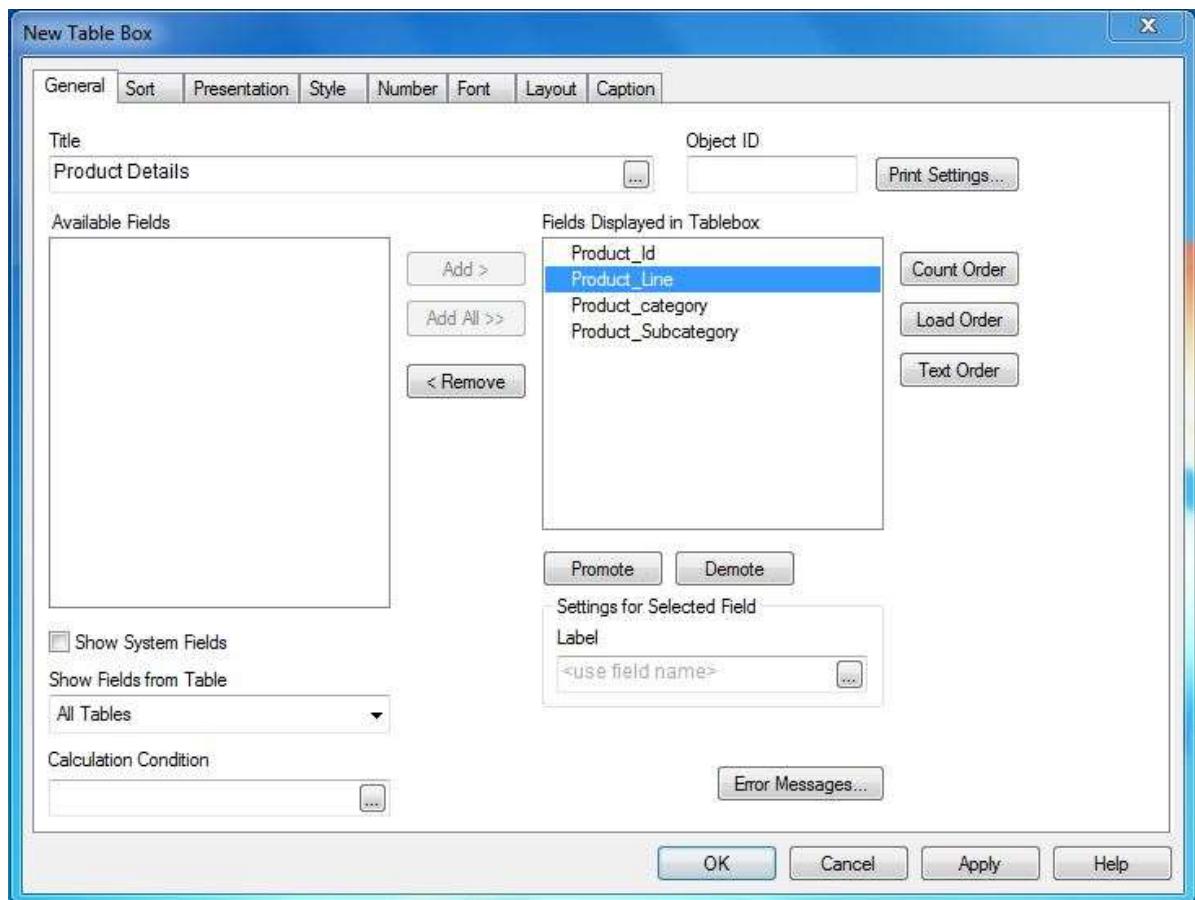
Now the script wizard prompts to save the file in the form of \*.qvw file extension. It asks to select a location where you need to save the file. Click "Next step" to proceed. Now, it is time to see the data that is loaded from the delimited file. We use a **Table Box** sheet object to display this data.

## Create Table Box

The **Table Box** is a sheet object to display the available data as a table. It is invoked from the menu **Layout -> New Sheet Object -> Table Box.**



On clicking Next, we get the option to choose the fields from the Table Box. You can use the **Promote or Demote** buttons to rearrange the fields.



## Table Box Data

On completing the above step, the Table Box Sheet Object appears which shows the data that is read from the Excel file.

The screenshot shows the QlikView interface with a window titled "QlikView x64 Personal Edition - [C:\Qlikview\files\Excel\using\_excel\_file.qvw]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. The toolbar below has various icons for file operations like Open, Save, Print, and search. The main area is titled "Main" and contains a table titled "Product Details". The table has four columns: Product\_Id, Product\_Line, Product\_category, and Product\_Subcategory. The data is as follows:

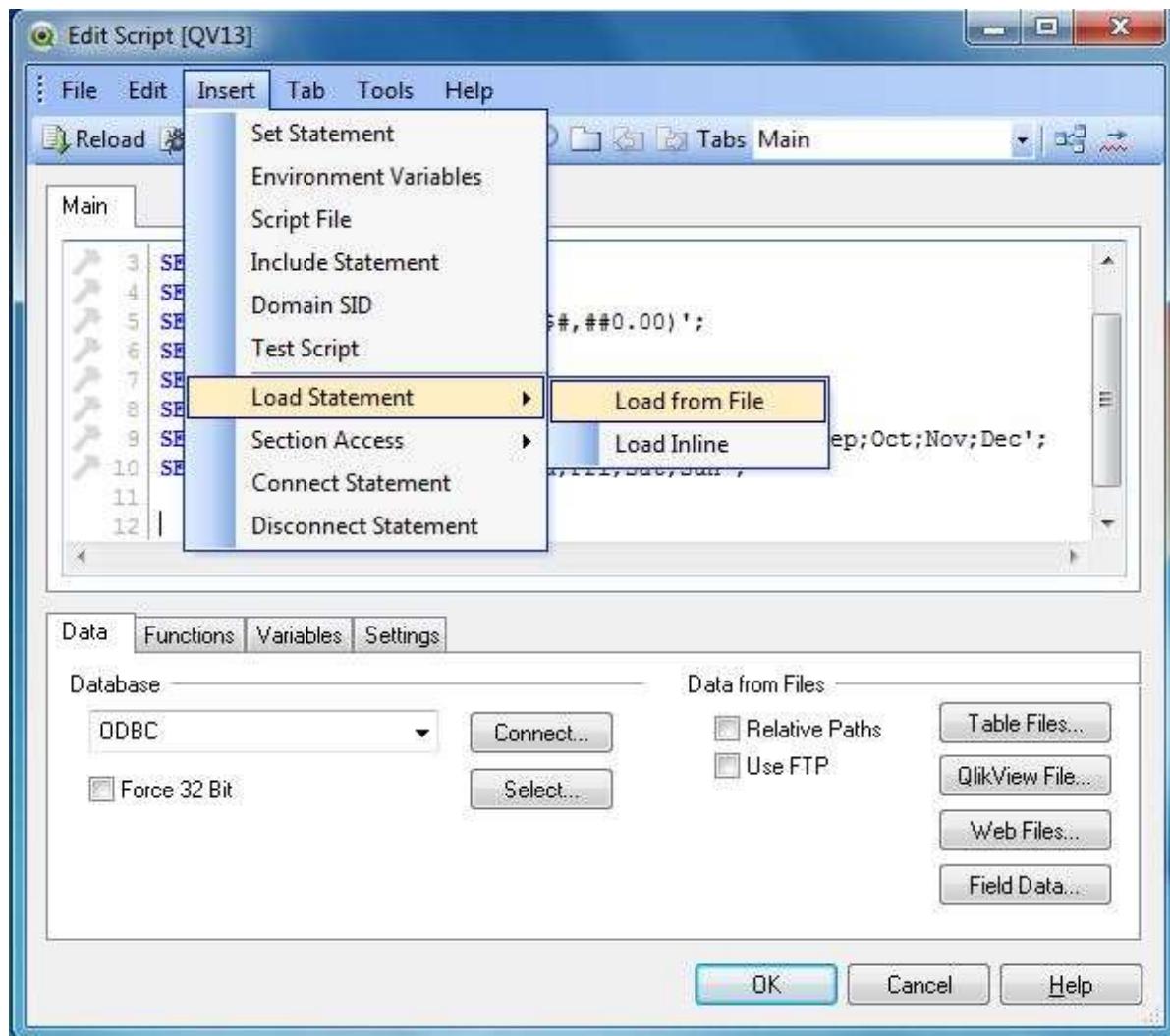
Product_Id	Product_Line	Product_category	Product_Subcategory
1	Sporting Goods	Outdoor Recreation	Winter Sports & Activities
2	Food, Beverages & Tobacco	Food Items	Fruits & Vegetables
3	Apparel & Accessories	Clothing	Uniforms
4	Sporting Goods	Athletics	Rugby
5	Health & Beauty	Personal Care	-
6	Arts & Entertainment	Hobbies & Creative Arts	Musical Instruments
7	Arts & Entertainment	Hobbies & Creative Arts	Orchestra Accessories
8	Arts & Entertainment	Hobbies & Creative Arts	Crafting Materials
9	Hardware	Tool Accessories	Power Tool Batteries
10	Home & Garden	Bathroom Accessories	Bath Caddies
11	Food, Beverages & Tobacco	Food Items	Frozen Vegetables
12	Home & Garden	Lawn & Garden	Power Equipment

At the bottom left, it says "For Help, press F1". At the bottom right, it shows the date and time "11/14/2015 9:37:09 AM" and the size "12 X 4".

## 7. QlikView – XML File

XML is a file format, which shares both the file format and the data on the World Wide Web, intranets, and elsewhere using standard ASCII text. It stands for Extensible Markup Language (XML). Similar to HTML it contains markup tags. However, unlike HTML where the markup tag describes structure of the page, in XML the markup tags describe the meaning of the data contained into the file. QlikView can use the data from XML files.

The process to load the data from XML files is similar to the loading of delimited files we have seen earlier. Open the script editor. Click on the menu **Insert -> Load Statement -> Load from File**. Browse for the XML file you wish to load. In this example, we are choosing the employee\_dat.xml file.



Select the XML File Structure

35

On opening the selected XML file, a window comes up as shown below. Under the **File Type** section in the left, choose XML. The content of the XML file now appears as a table along with the header column. Click Finish.

File Wizard : Type

File Type

- Delimited
- Fixed Record
- Dif
- Excel (xls)
- Excel (xlsx)
- Html
- Qvd
- Xml
- Qvx

Tables XML

RECORDS/EMPLOYEE

Fields

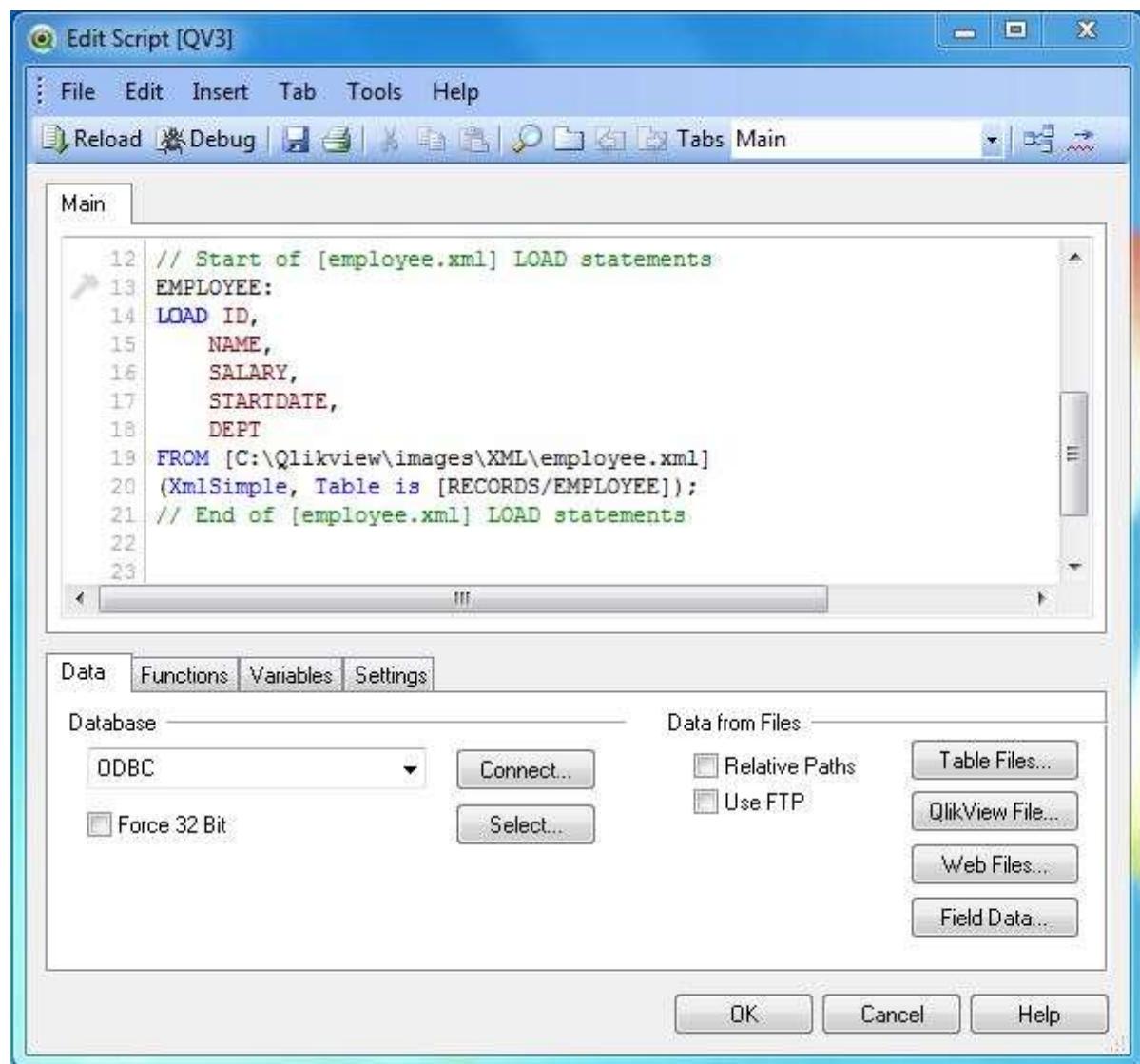
DEPT (8 values)  
ID (8 values)  
NAME (8 values)  
SALARY (8 values)  
STARTDATE (8 values)

ID	NAME	SALARY	STARTDATE	DEPT
1	Rick	623.3	1/1/2012	IT
2	Dan	515.2	9/23/2013	Operations
3	Michelle	611	11/15/2014	IT
4	Ryan	729	5/11/2014	HR
5	Gary	843.25	3/27/2015	Finance
6	Drensh	570	5/21/2013	IT

< Back    Next >    Finish    Cancel    Help

## File Loader Script

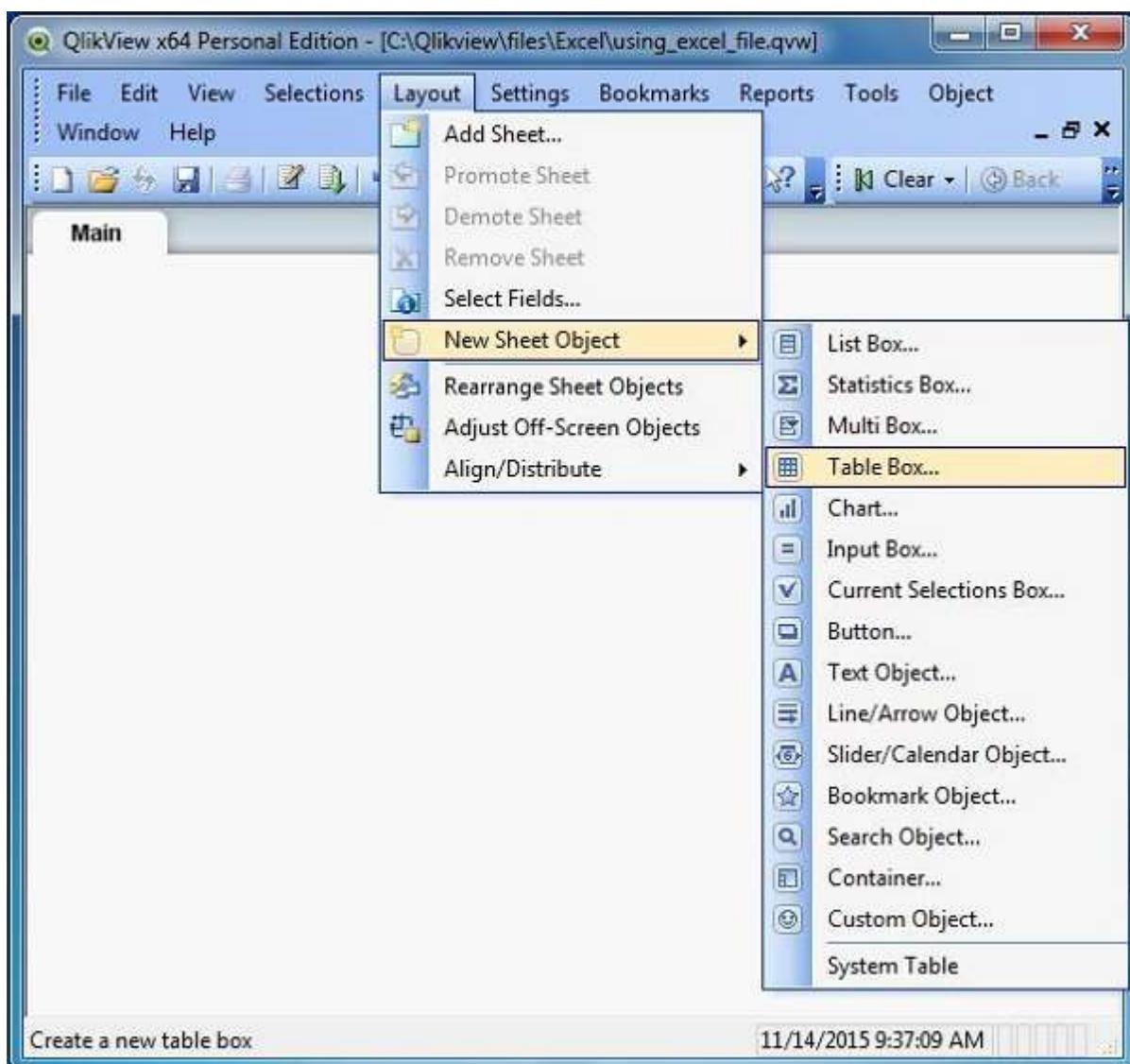
The loading of the XML file into QlikView is done through the load script, which can be seen below. So when we use any XML file, we can tweak the below given script to rename the columns or change the file location etc.



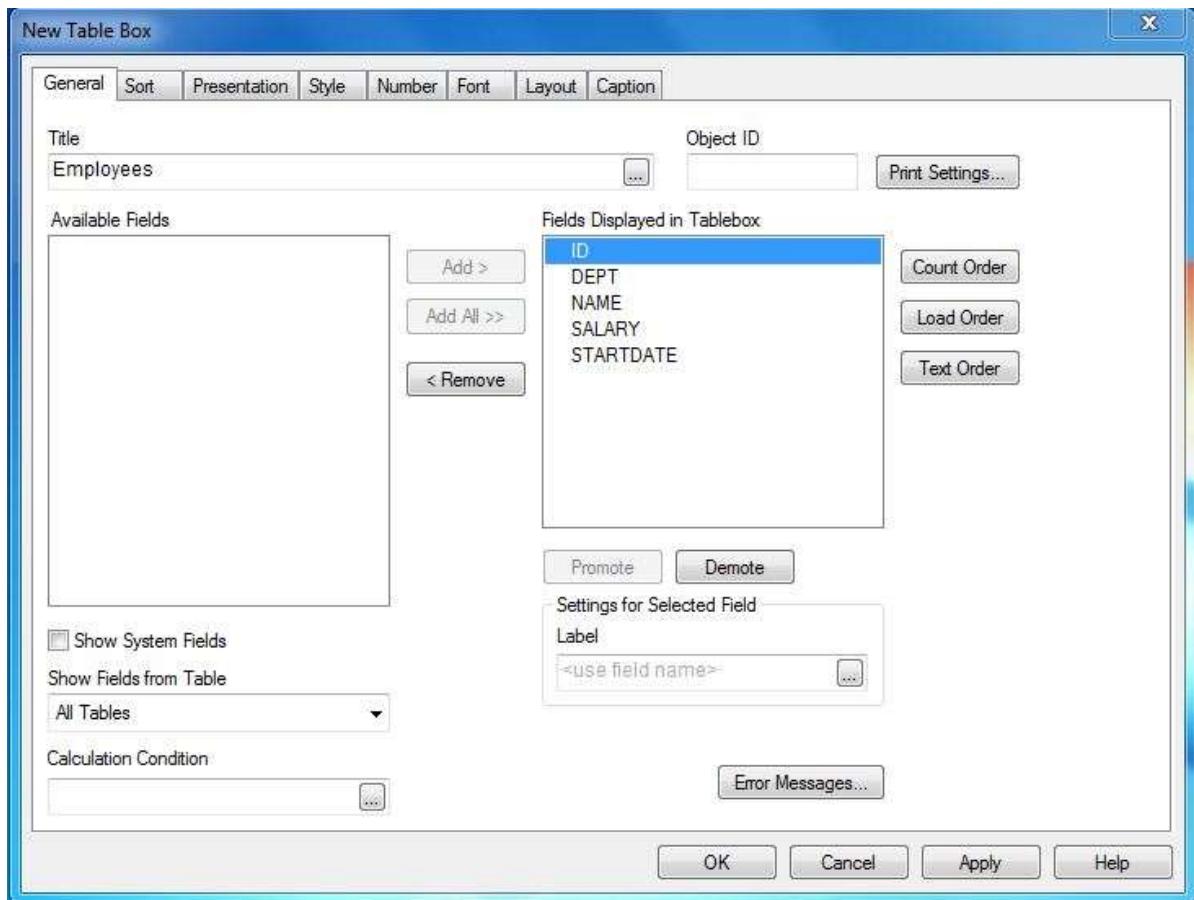
Now the script wizard prompts you to save the file in the form of \*.qvw file extension. It asks to select a location where you need to save the file. Click "Next step" to proceed. Now it is time to see the data that is loaded from the XML file. We use a **Table Box** sheet object to display this data.

## Create Table Box

The **Table Box** is a sheet object to display the available data as a table. It is invoked from the menu **Layout -> New Sheet Object -> Table Box**.



On clicking Next, we get the option to choose the fields from the Table Box. You can use the **Promote or Demote** buttons to rearrange the fields.



## Table Box Data

On completing the above step, the Table Box Sheet Object appears which shows the data that is read from the Excel file.

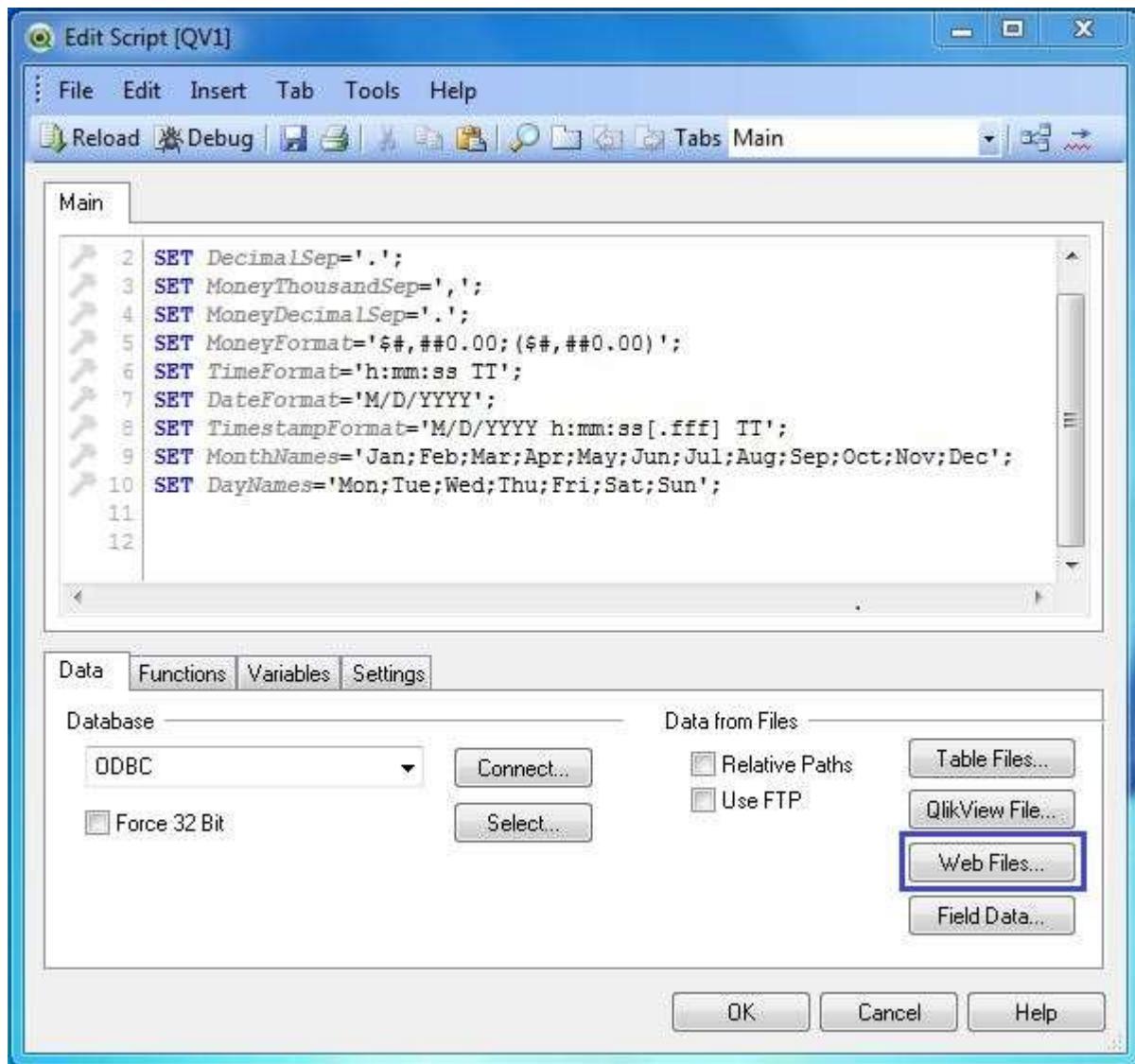
The screenshot shows the QlikView interface with a window titled "QlikView x64 Personal Edition - [C:\Qlikview\datafiles\xml\_file.qvw]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. The toolbar below has various icons for file operations like Open, Save, Print, and search. The main area is labeled "Main" and contains a table titled "Employees". The table has columns: ID, DEPT, NAME, SALARY, and STARTDATE. The data is as follows:

ID	DEPT	NAME	SALARY	STARTDATE
1	IT	Rick	623.3	1/1/2012
2	Operations	Dan	515.2	9/23/2013
3	IT	Michelle	611	11/15/2014
4	HR	Ryan	729	5/11/2014
5	Finance	Gary	843.25	3/27/2015
6	IT	Pranab	578	5/21/2013
7	Operations	Tusar	632.8	7/30/2013
8	Finance	Rasmi	722.5	6/17/2014

At the bottom, there is a status bar with "For Help, press F1", the date and time "11/14/2015 5:17:22 PM", and a resolution indicator "8 X 5".

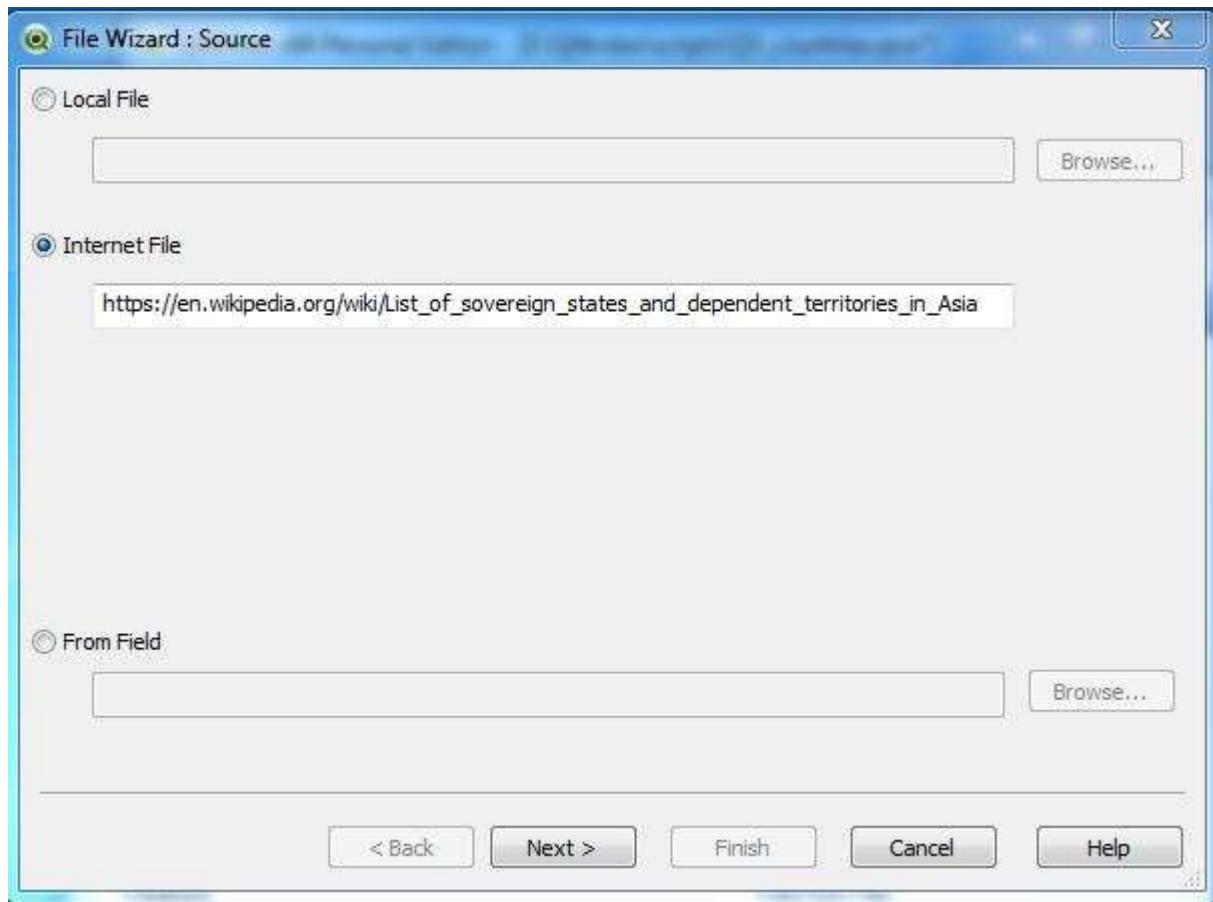
## 8. QlikView – Web File

QlikView can process files from the web, which are in the HTML format. It can extract data from HTML tables. The URL of the web file to be processed is given as an input and QlikView fetches both, the structure and content of the file. Then it analyzes the structure of the page extracting the relevant data from the HTML tables present in the page. We choose the **Web files** option from the **Data from files** section under the Data tab of Script Editor.



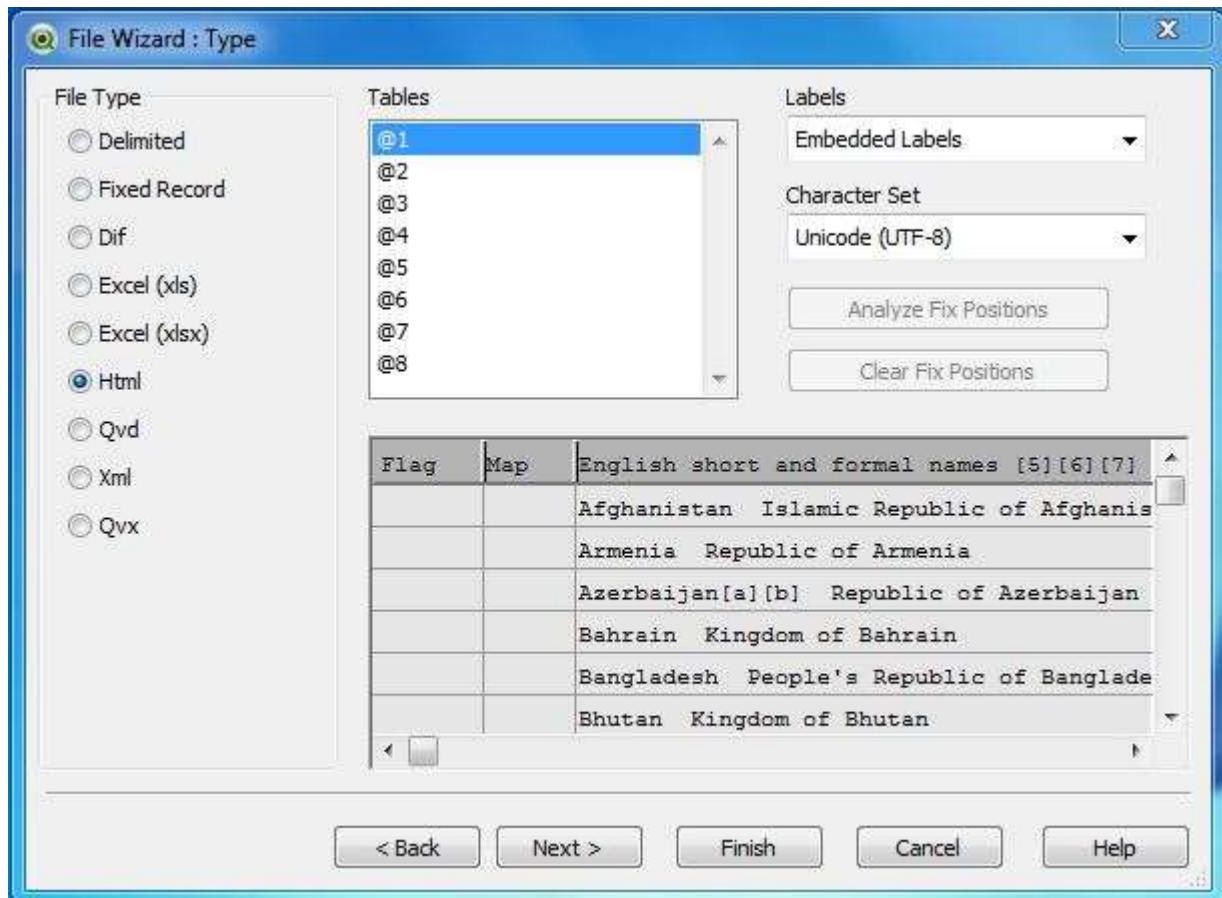
## Give the URL as Input

On selecting the Web files option, we get a new window to give the URL as input. In this example, we are choosing the List of sovereign states and dependent territories in Asia as the input page from Wikipedia. Mention the URL and click Next.



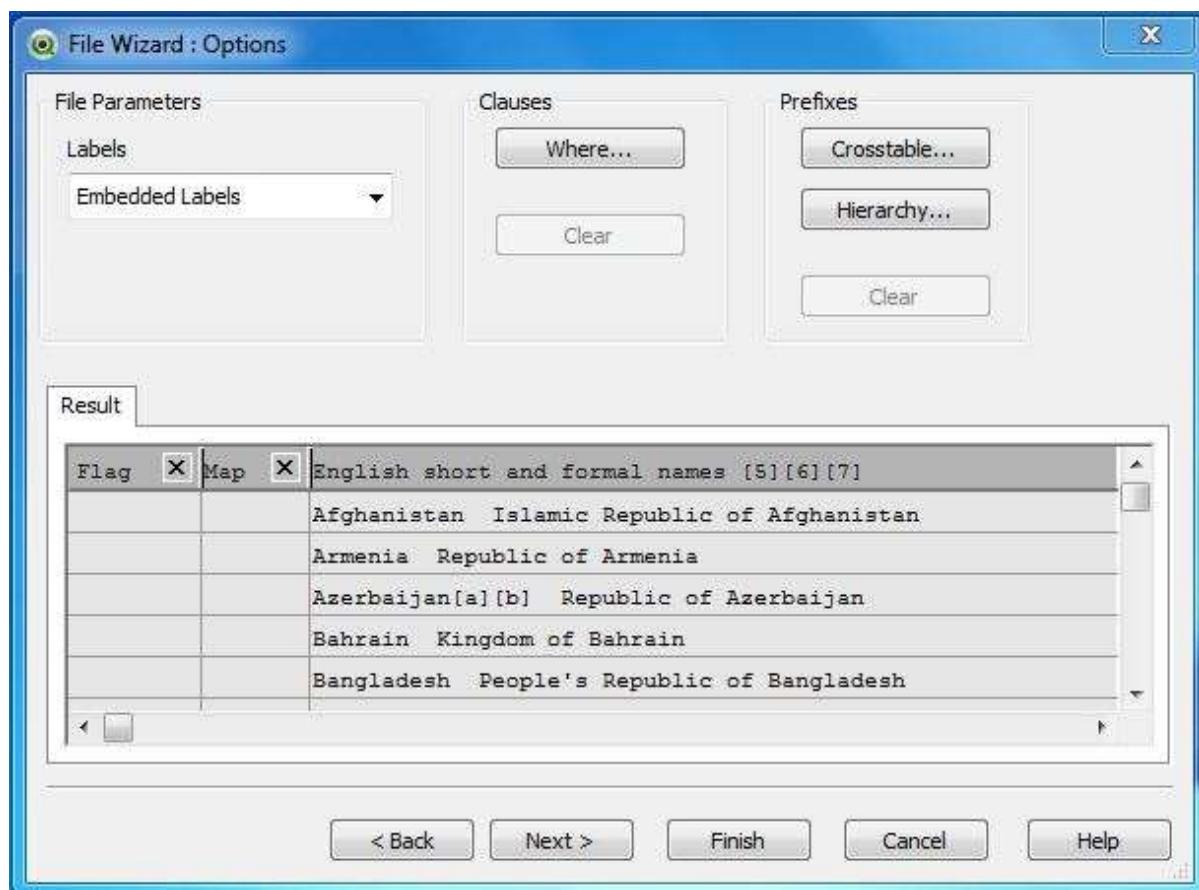
## Select the Table from the Web File

On opening the selected Web file, the window shown below comes up. Here we can see the various tables present in the webpage labeled as @1, @1, @3 and so on. Choose the first table and click Next twice.



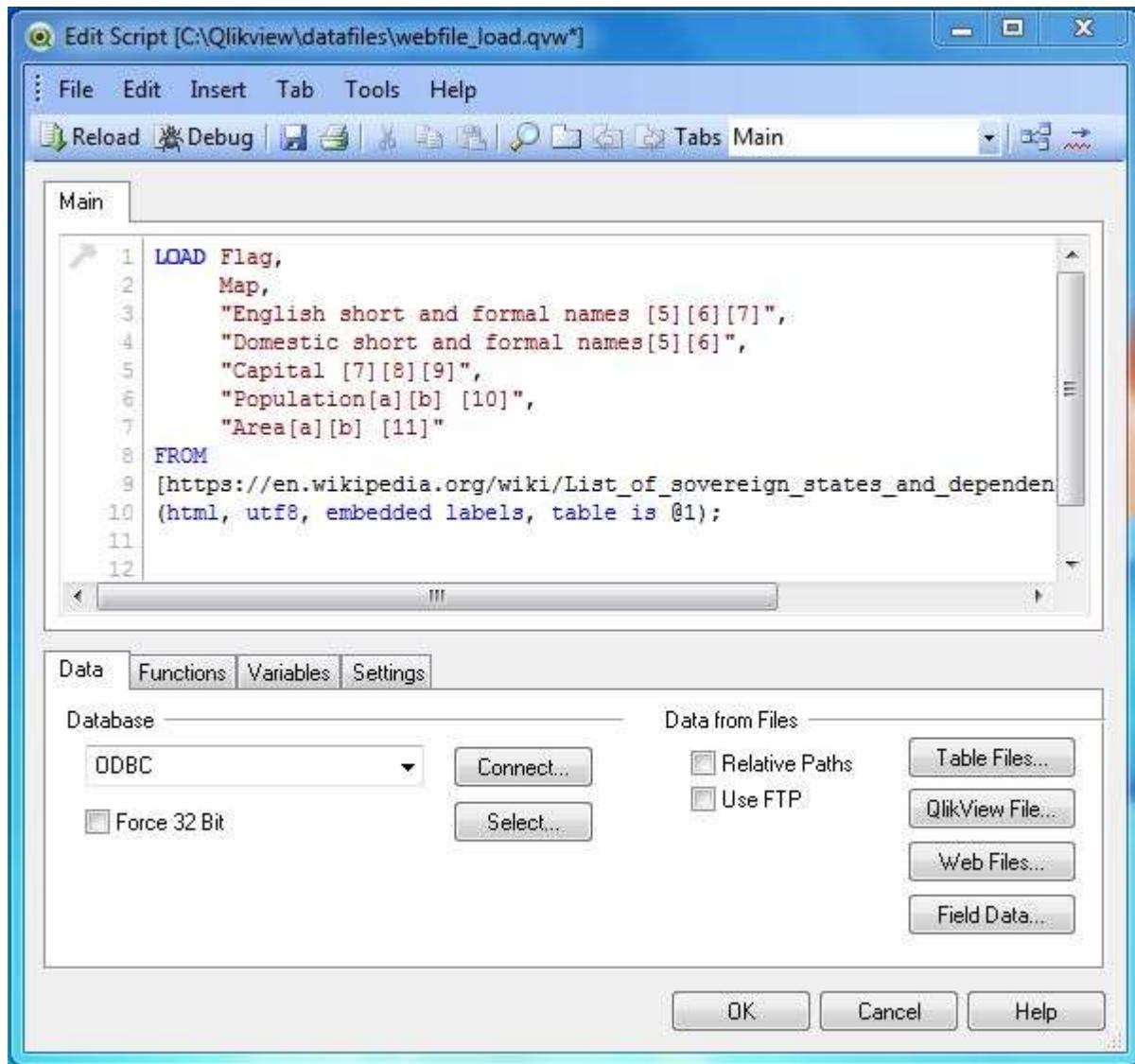
## Select the Columns of the Table

From the above table, we can choose only the columns we need by removing the unwanted columns using the cross sign.



## Load Script

The loading of the file into QlikView is done through the load script, which can be seen in the screen shot given below. Hence, when we use any delimited file, we can tweak the below given script as per the file format.



The screenshot shows the 'Edit Script' dialog box in QlikView. The title bar says 'Edit Script [C:\Qlikview\datafiles\webfile\_load.qvw]'. The menu bar includes File, Edit, Insert, Tab, Tools, Help. The toolbar has icons for Reload, Debug, Save, Print, Find, Copy, Paste, and Tabs. The tabs at the top are Main (selected), Data, Functions, Variables, and Settings. The main area contains a script editor with the following code:

```

1 LOAD Flag,
2 Map,
3 "English short and formal names [5][6][7]",
4 "Domestic short and formal names[5][6]",
5 "Capital [7][8][9]",
6 "Population[a][b] [10]",
7 "Area[a][b] [11]"
8 FROM
9 [https://en.wikipedia.org/wiki/List_of_sovereign_states_and_dependen
10 (html, utf8, embedded labels, table is @1);
11
12

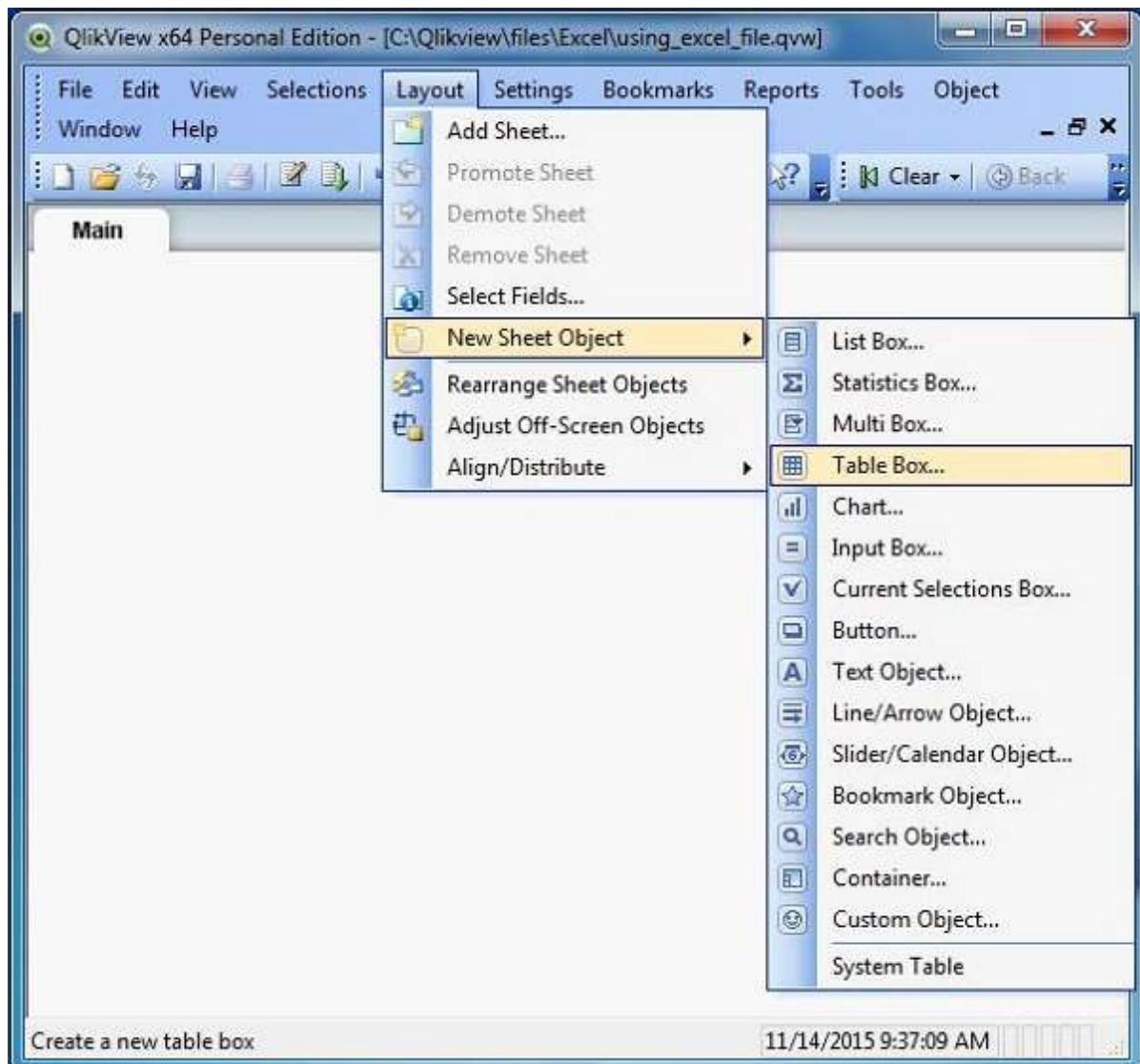
```

Below the script editor are two sections: 'Database' and 'Data from Files'. The 'Database' section has dropdown menus for 'ODBC' and 'Select...', and checkboxes for 'Force 32 Bit' and 'Table Files...'. The 'Data from Files' section has checkboxes for 'Relative Paths' and 'Use FTP', and buttons for 'Table Files...', 'QlikView File...', 'Web Files...', and 'Field Data...'. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

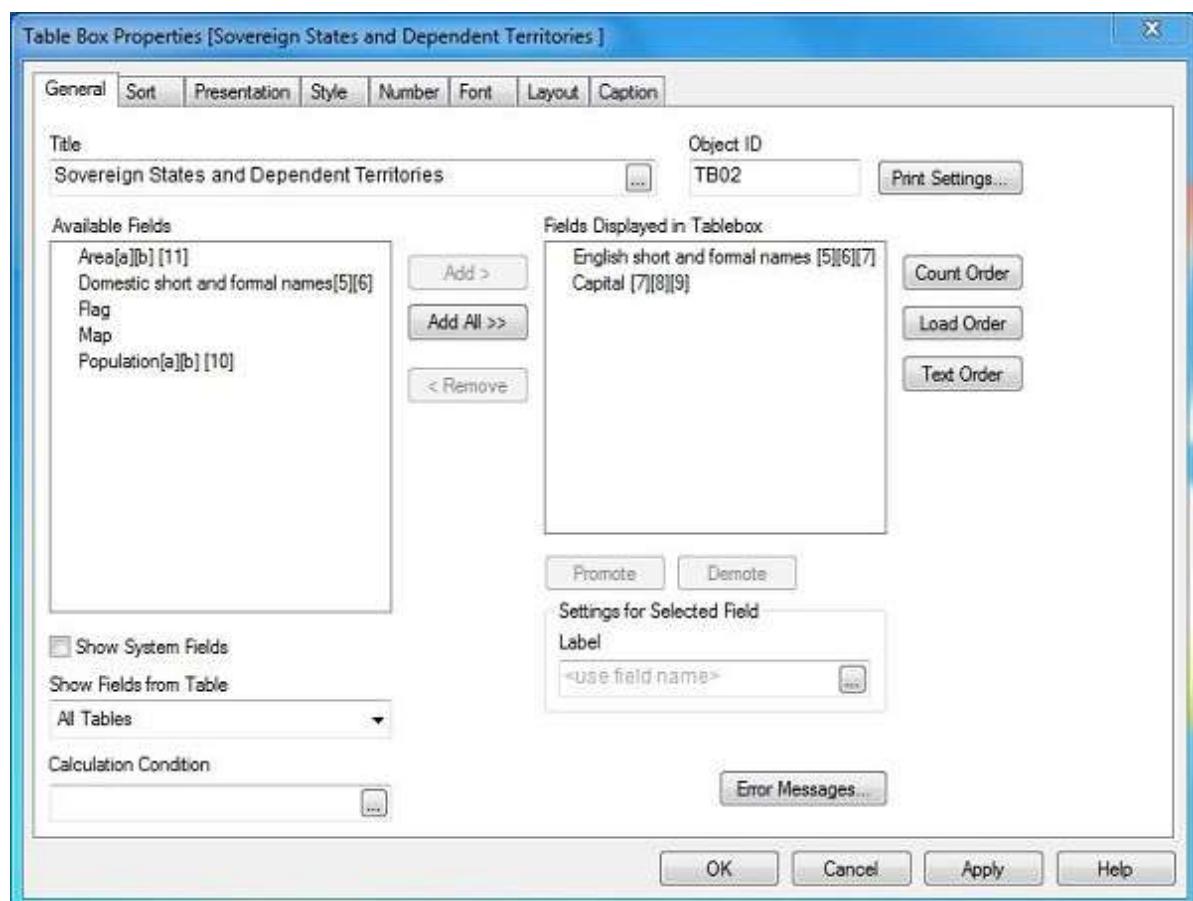
Now the script wizard prompts to save the file in the form of \*.qvw file extension. It asks to select a location where you need to save the file. Click "Next step" to proceed. Now it is time to see the data that is loaded from the web file. We use a **Table Box** sheet object to display this data.

## Create Table Box

The **Table Box** is a sheet object to display the available data as a table. It is invoked from the menu **Layout -> New Sheet Object -> Table Box**.



On clicking Next, we get the option to choose the fields from the Table Box. You can use the **Promote or Demote** buttons to rearrange the fields.



## Table Box Data

On completing the above step, the Table Box Sheet Object appears, which shows the data that is read from the Web file. Mark the **Non-English** characters!!

English short and formal names [5][6]...	Capital [7][8][9]
Afghanistan Islamic Republic of Afghanistan	Kabul Dari: کابل (Kābul) Pashto: کابل (Kābul)
Armenia Republic of Armenia	Yerevan Armenian: Երևան (Yerevan)
Azerbaijan[a][b] Republic of Azerbaijan	Baku Azerbaijani: Bakı
Bahrain Kingdom of Bahrain	Manama Arabic: مانامه (Al Manāmah)
Bangladesh People's Republic of Bangladesh	Dhaka Bengali: ঢাকা (Dhākā)
Bhutan Kingdom of Bhutan	Thimphu Dzongkha: གྲୟମྴସ (Thimphu)
Brunei Brunei Darussalam	Bandar Seri Begawan English: Bandar Seri Begawan Malay: B
Cambodia Kingdom of Cambodia	Phnom Penh Khmer: ភ្នំពេញ (Phnum Pénh)
China[b] People's Republic of China	Beijing Chinese: 北京 (Beijing)
Cyprus[b] Republic of Cyprus	Nicosia Greek: Λευκωσία (Lefkosía) Turkish: Lefkoşa
Georgia[a][b]	Tbilisi / T'bilisi Georgian: თბილისი (Tbilisi)
India Republic of India	New Delhi English: New Delhi Hindi: नई दिल्ली (Nai Dilli)
Indonesia[a] Republic of Indonesia	Jakarta Indonesian: Jakarta
Iran Islamic Republic of Iran	Tehran Persian: تهران (Tehrān)
Iraq Republic of Iraq	Baghdad Arabic: بغداد (Baghdād)
Israel State of Israel	Jerusalem (Claimed and de facto)[c] Hebrew: יְרוּשָׁלָם (Yeru

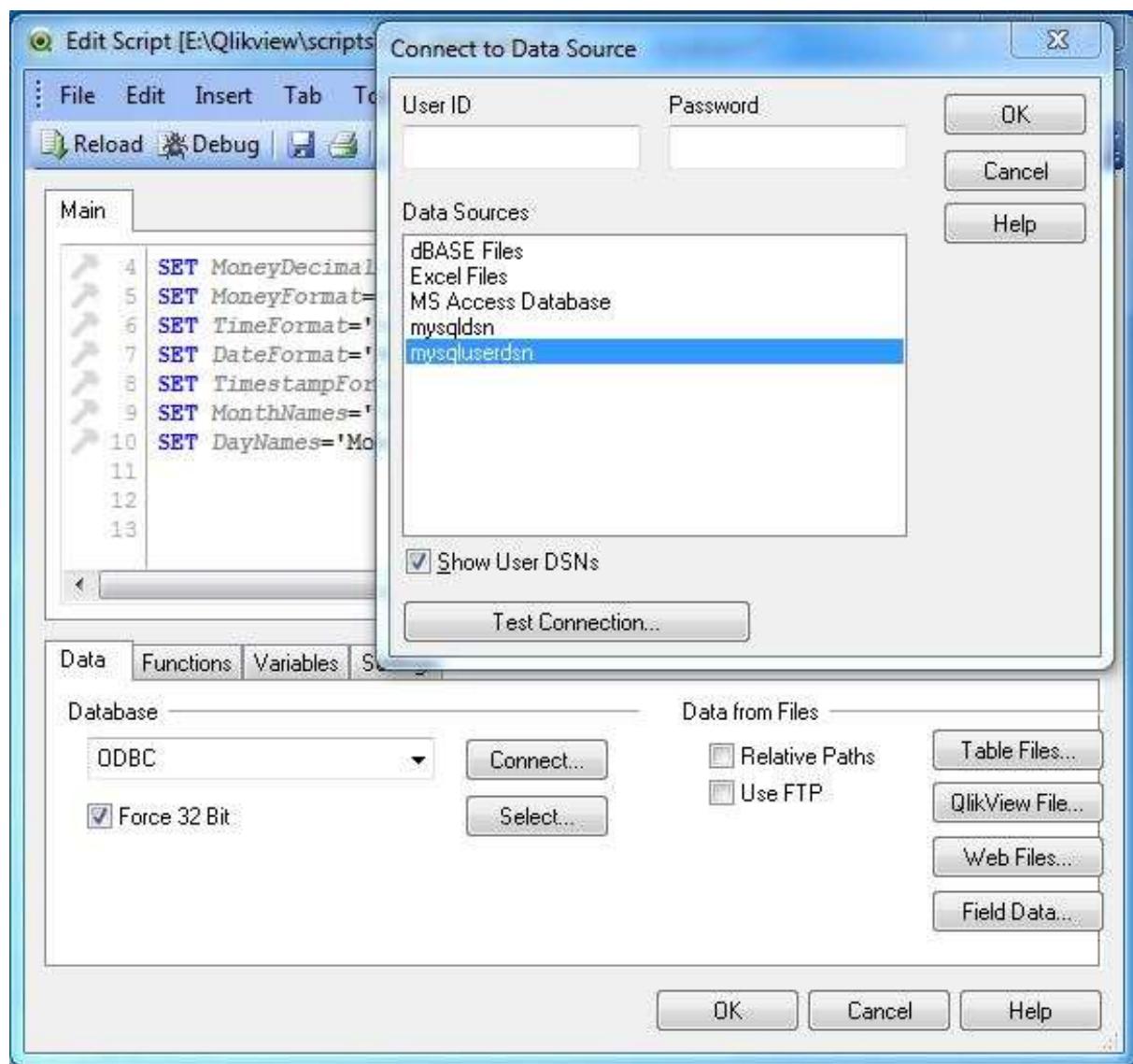
## 9. QlikView – Database Connection

QlikView can connect to most of the popular databases like MySQL, SQL Server, Oracle, Postgress etc. It can fetch data and table structures into QlikView environment and store the results in its memory for further analysis. The steps to connect to any of these databases involves creating an ODBC connection using a DSN and then using this DSN to fetch the data.

For this tutorial, we will be connecting to MySQL database. This tutorial assumes you have a MySQL environment available. Create an ODBC DSN (Data Source Name) for MySQL, following these steps - [to create DSN](#). Name the DSN as **mysqluserdsn** or you may prefer to use the existing one if you have already created a DSN for MySql.

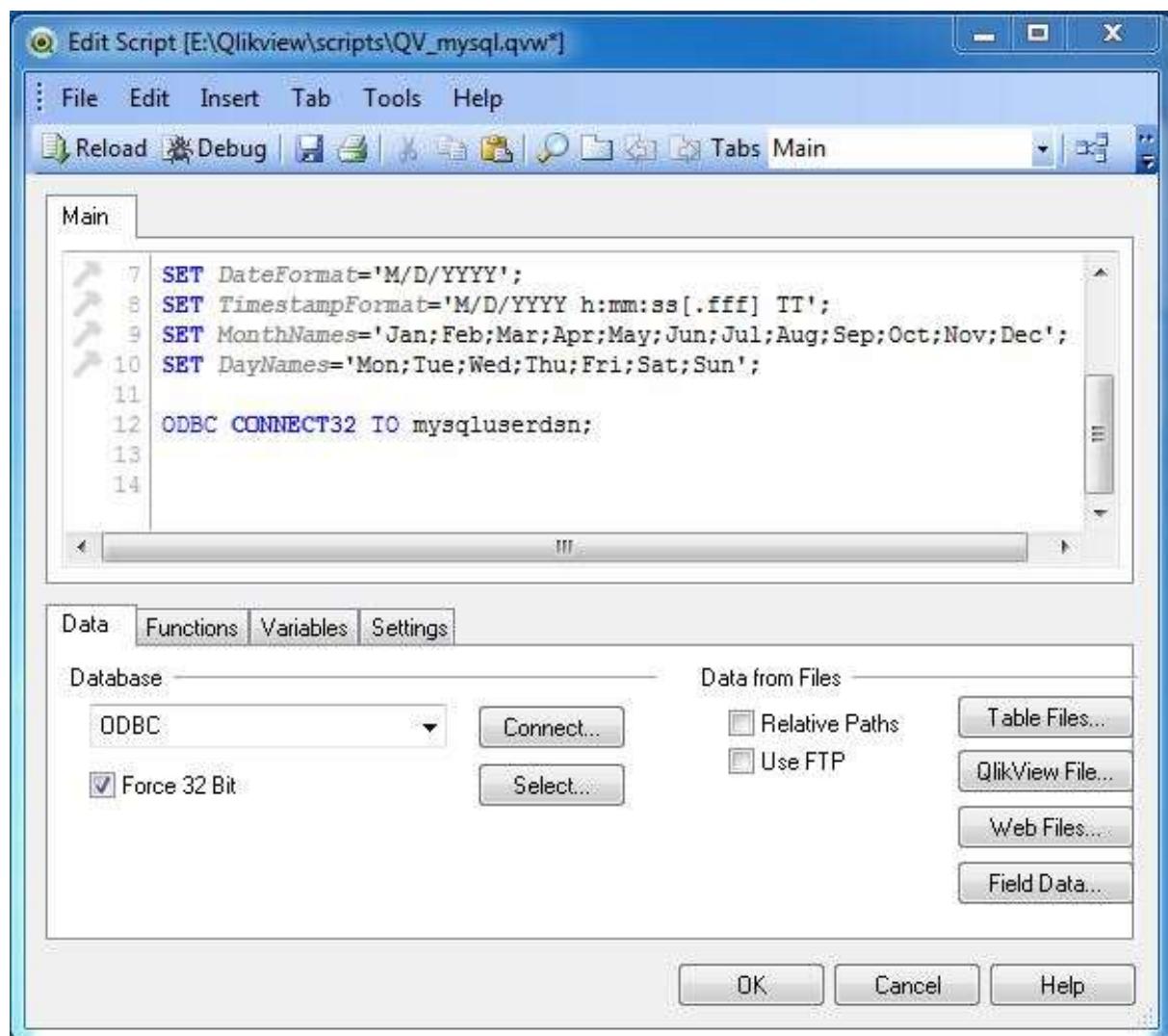
### Connecting to the Database

For this chapter, we will use the MySQL inbuilt database named **sakila**. We create a new QlikView document and open the script editor (pressing Control+E). Under the tab **Data**, we locate the section named Database. Choose ODBC from the drop down list and click Connect. The following window opens. Choose the DSN named **mysqluserdns** and click Test Connection. The message **Connection Test succeeded** should appear.



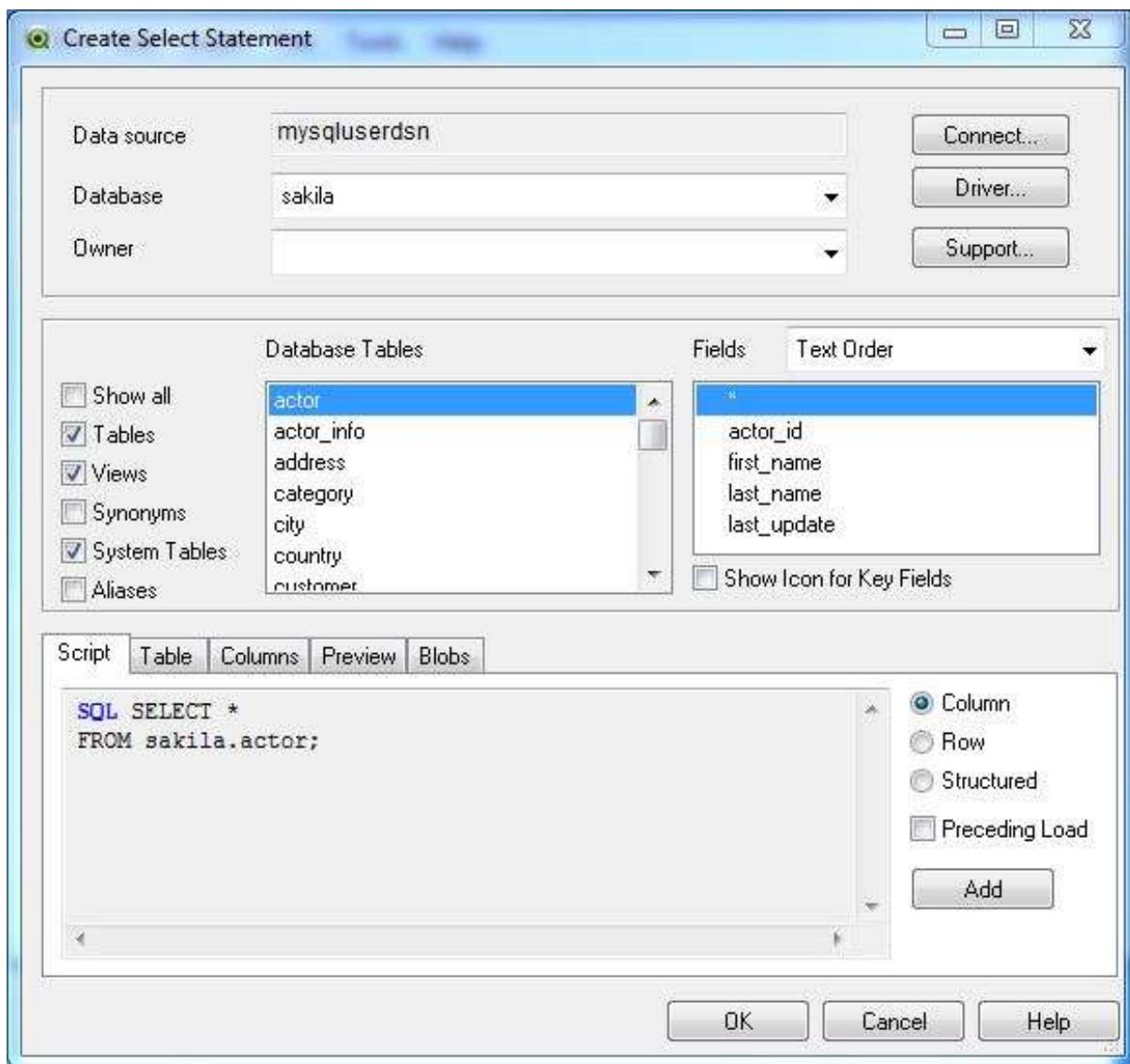
## Verifying Connection to Database

On successful connection, the screen given below appears showing the connection to the DB in the main window of the script editor.



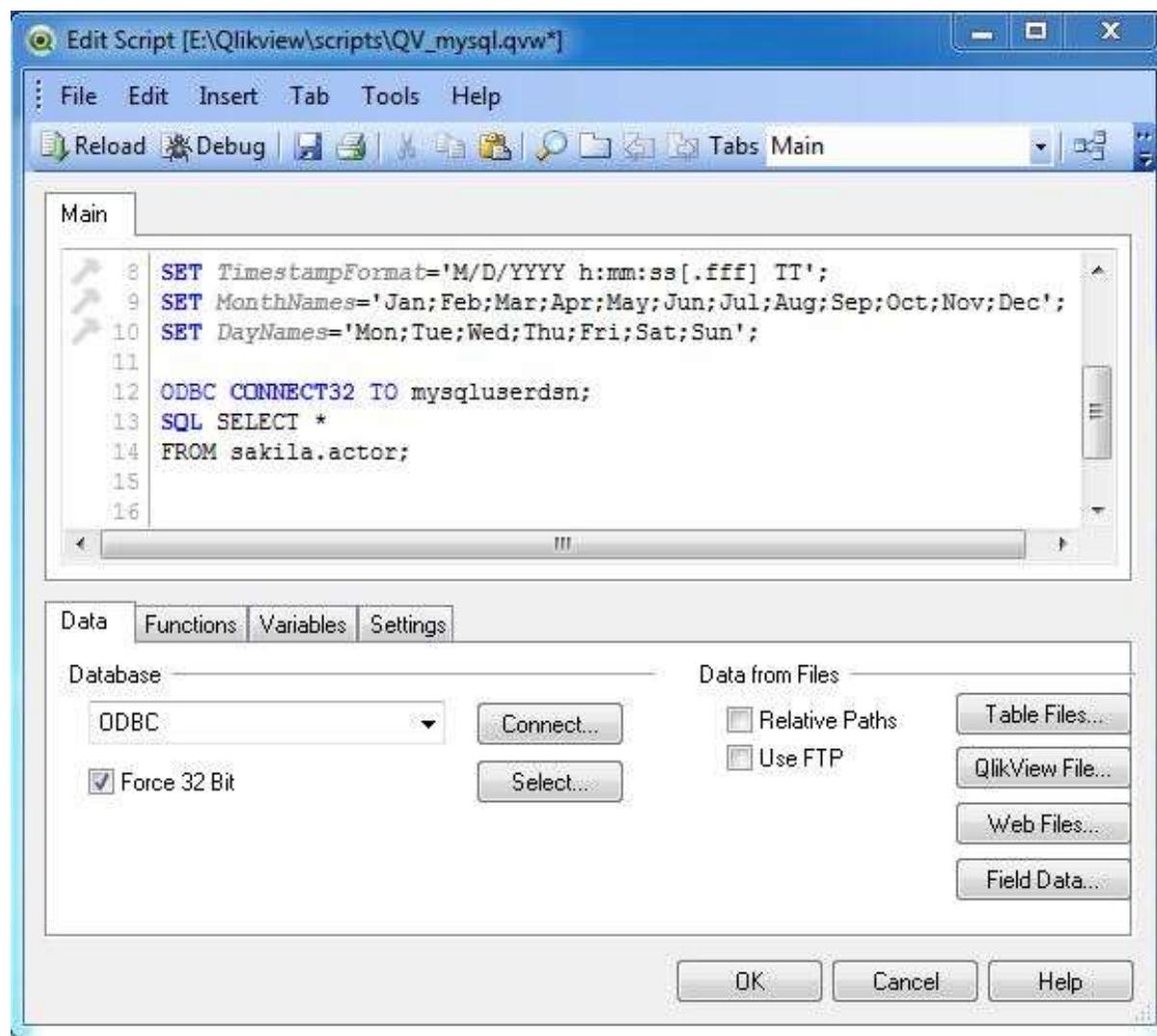
## Select Database Table

Click **Select** in the above window to get the list of tables and columns. Here as we have created the DSN with **sakila** as the default database we get the list of tables and columns from this database. We can choose another database from the database drop down list as shown in the screenshot given below. We will continue using the sakila database for this chapter.



## Table Loader Script

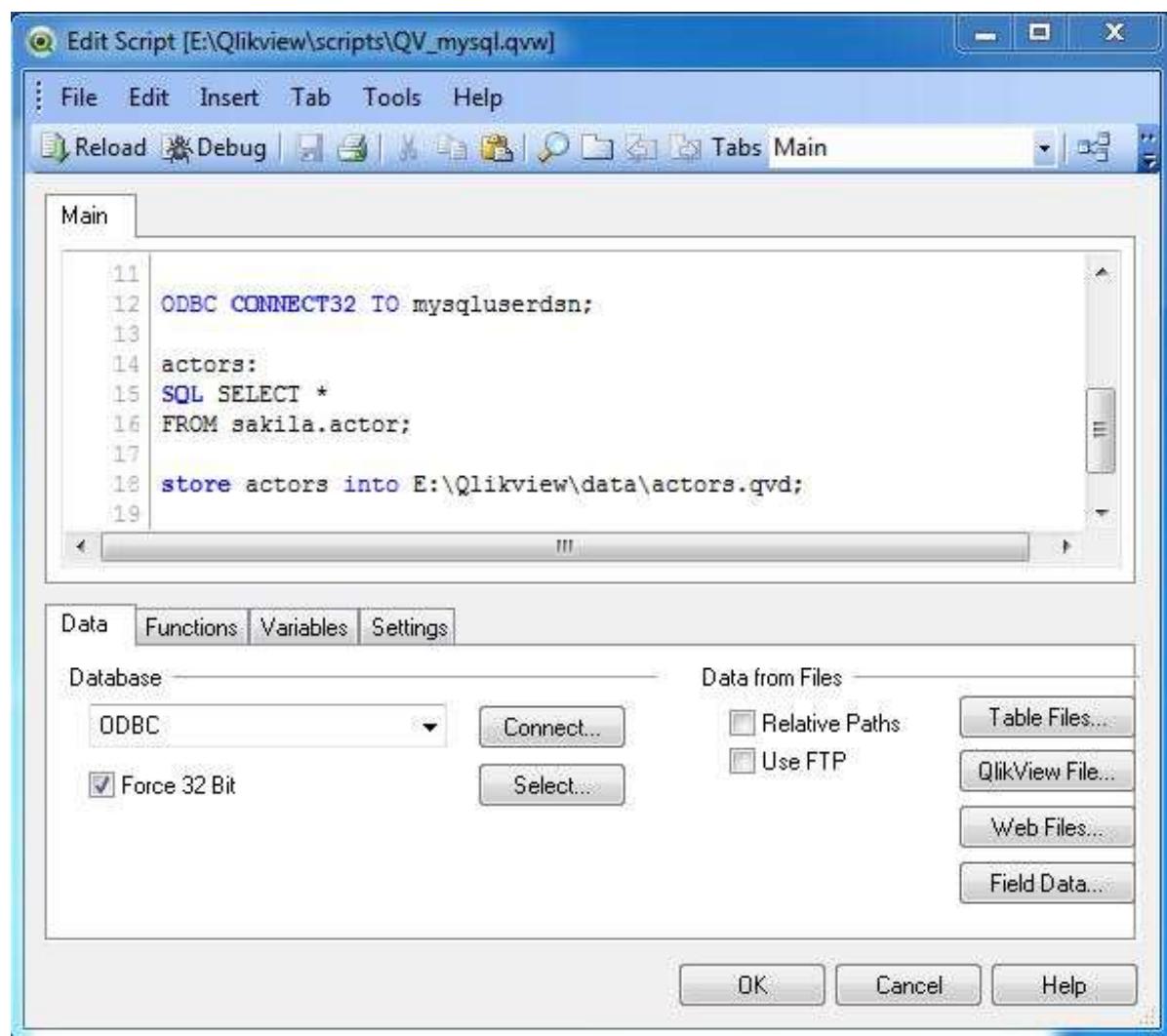
On Clicking OK in the above window, we get back to the main script editor showing the script for using the table named **actor**.



## Showing the Result in qvd File

Now the data loaded into QlikView document needs to be stored permanently to be analyzed further. For this, we will edit the script to store the data in the form of a qvd file. Press **Control+E** to open the edit script window and write the following code.

In the code, we give appropriate names to the columns and mention the table name above the load statement. In addition, we give a path where the generated qvd file will be stored. Save this file as **QV\_mysql.qvw**



## Using the qvd File

The qvd file can be loaded into the main document and used to create graphs and tables for further analysis. Press **Control+R** to reload the **QV\_mysql.qvw** file and click **Next** in the chart wizard. Choose the straight table to be created with `actor_id`, `first_name`, `last_name` as the dimensions and count of `actor_id` as the expression. A chart appears as given below.

QlikView x64 Personal Edition - [QV14]

Main

Count(actor_id)			
actor_id	first_name	last_name	Count(actor_id)
1	PENELOPE	GUINNESS	1
2	NICK	WAHLBERG	1
3	ED	CHASE	1
4	JENNIFER	DAVIS	1
5	JOHNNY	LOLLOBRIGIDA	1
6	BETTE	NICHOLSON	1
7	GRACE	MOSTEL	1
8	MATTHEW	JOHANSSON	1
9	JOE	SWANK	1
10	CHRISTIAN	GABLE	1
11	ZERO	CAGE	1
12	KARL	BERRY	1
13	UMA	WOOD	1
14	VIVIEN	BERGEN	1
15	CUBA	OLIVIER	1
16	FRED	COSTNER	1

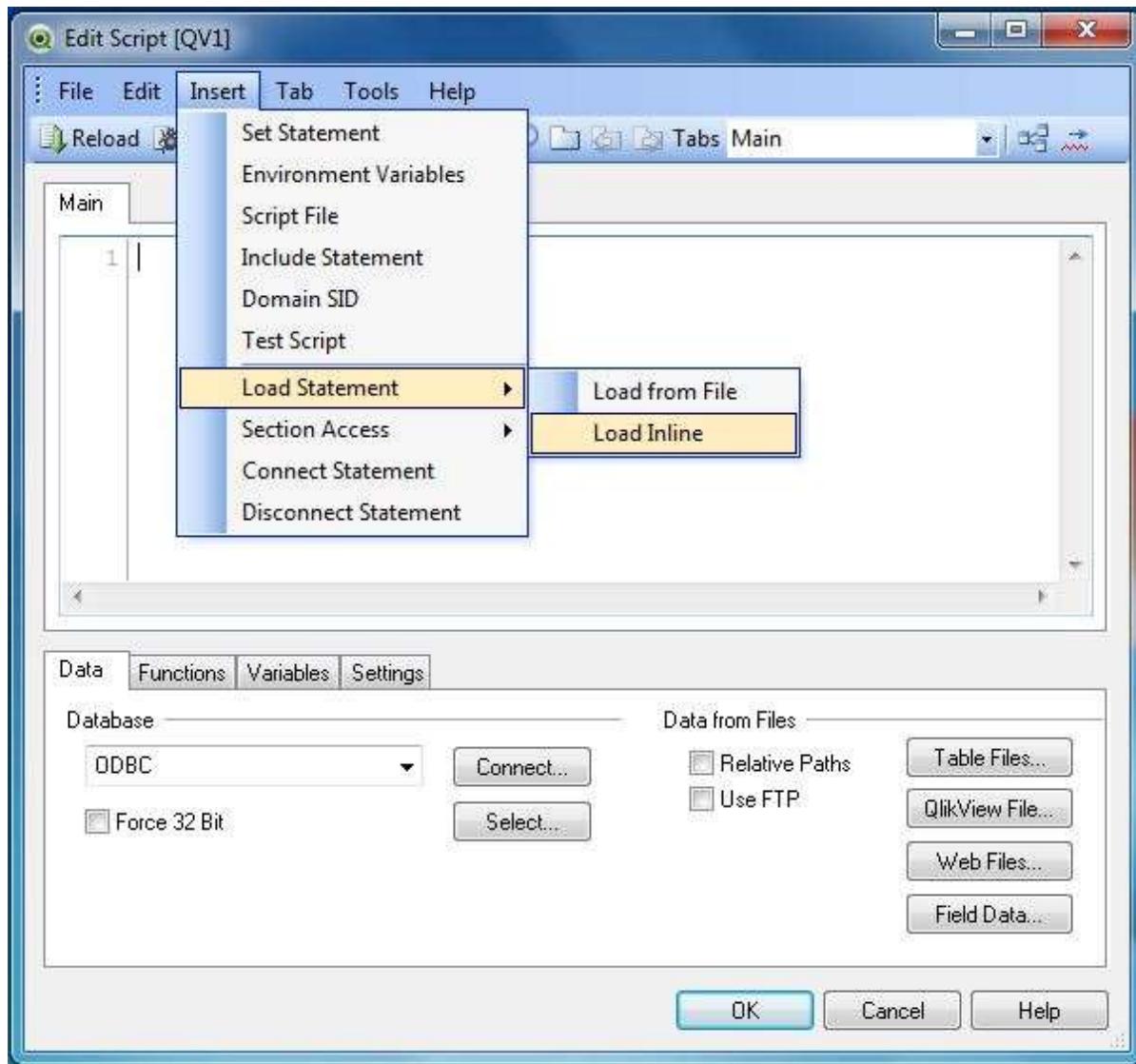
For Help, press F1

## 10. QlikView – Inline Data

Data can be entered into a QlikView document by directly typing or pasting it. This feature is a quick method to get the data from the clipboard into the QlikView. The script editor provides this feature under the Insert tab.

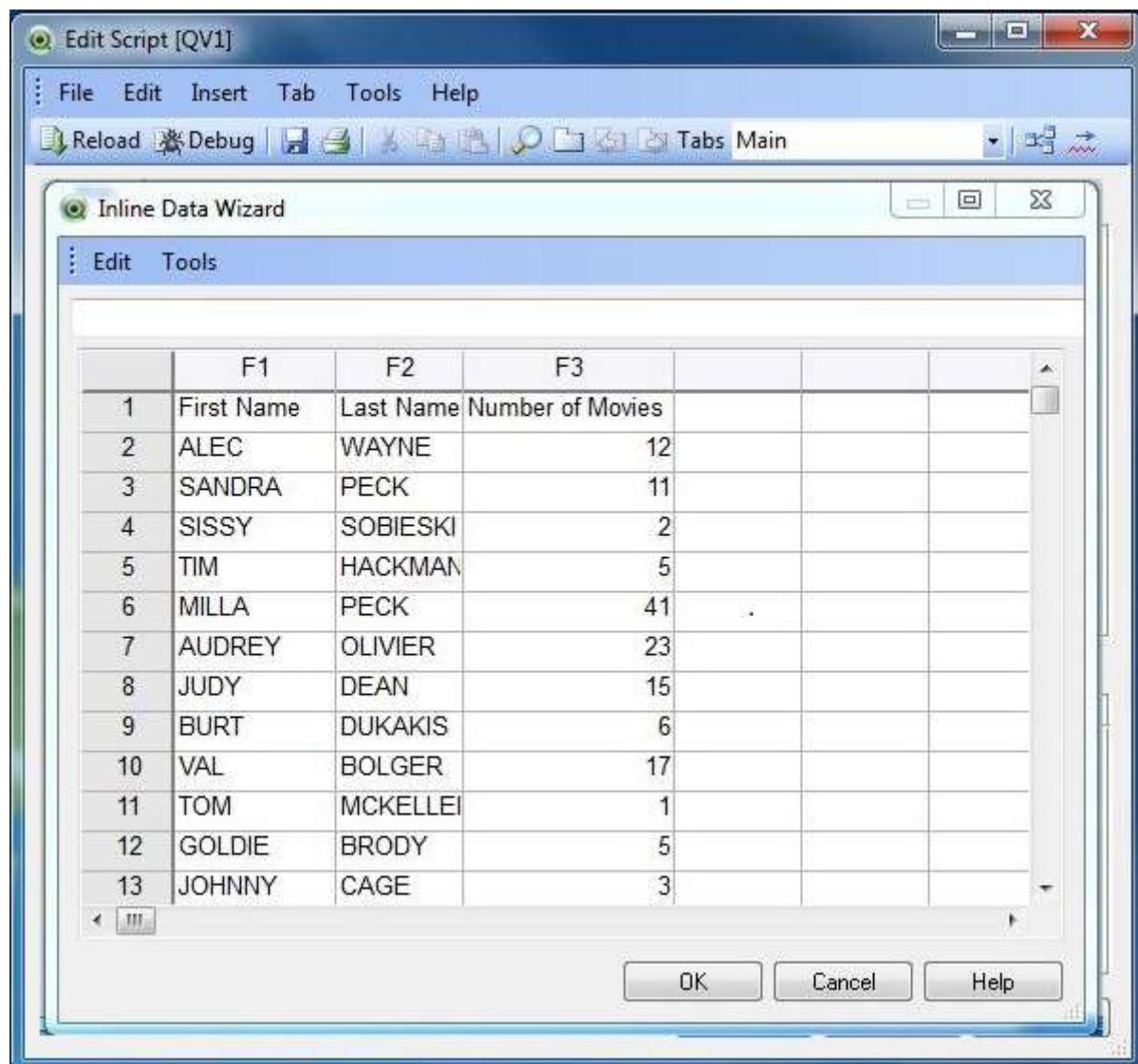
### Script Editor

To open the Inline data load option, we open the script editor and go to **Insert -> Load Statement -> Load Inline**.



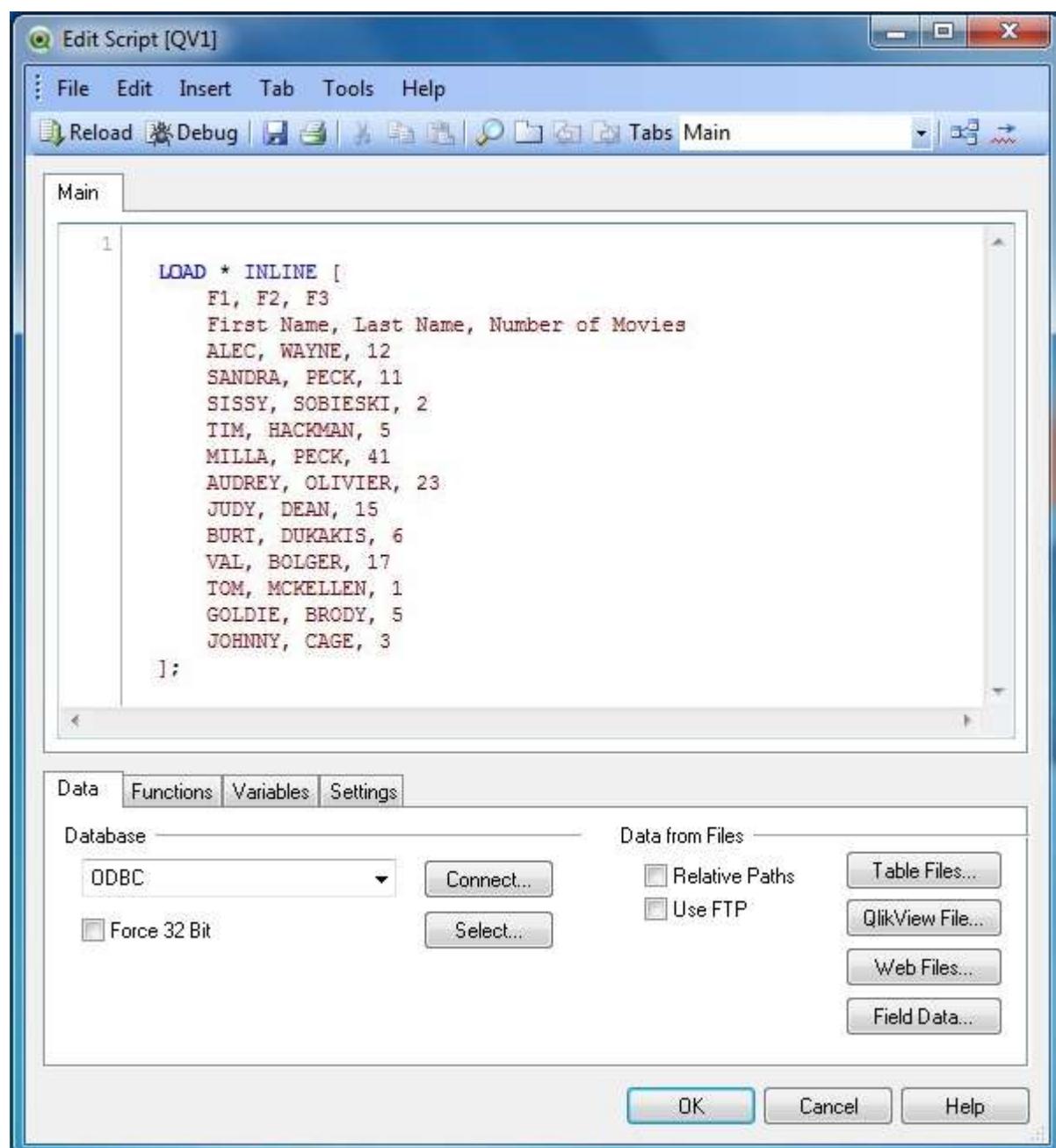
## Inserting Data

On opening the above screen, we get a spreadsheet-like document where we can type the values. We can also paste the values already available in the clipboard. Please note the column headers are created automatically. Click Finish.



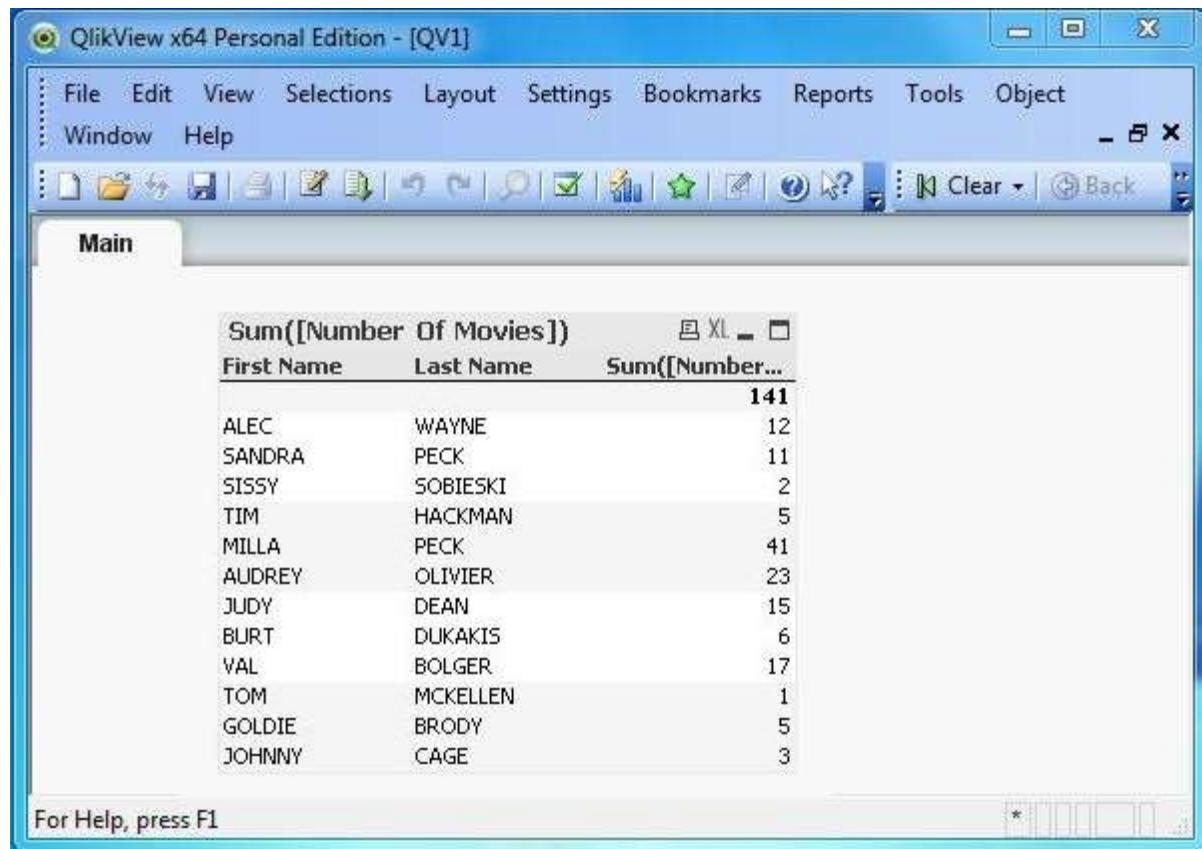
## Load Script

The command, which loads the data, is created in the background which can be seen in the script editor.



## Table Box Data

On creating a Table Box Sheet Object, we see the data that is read from the Inline data load option.



The screenshot shows the QlikView x64 Personal Edition interface with a window titled "QlikView x64 Personal Edition - [QV1]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. Below the menu is a toolbar with various icons. The main area is labeled "Main" and contains a table box. The table has three columns: "First Name", "Last Name", and "Sum([Number Of Movies])". The data is as follows:

First Name	Last Name	Sum([Number Of Movies])
ALEC	WAYNE	12
SANDRA	PECK	11
SISSY	SOBIESKI	2
TIM	HACKMAN	5
MILLA	PECK	41
AUDREY	OLIVIER	23
JUDY	DEAN	15
BURT	DUKAKIS	6
VAL	BOLGER	17
TOM	MCKELLEN	1
GOLDIE	BRODY	5
JOHNNY	CAGE	3

At the bottom left of the main area, it says "For Help, press F1".

# 11. QlikView – Resident Load

QlikView can load data from tables already existing in its RAM, which is already processed by a script. This requirement arises when you want to create a table deriving data from an already existing table in the same script. Please note that both the new table and the existing table should be in the same script.

## Creating the Load Script

Open the script editor (or use Control+E) and mention the following script. Here we create an inline table named **Regions** with sales data for different regions. Then we create another table named **Total** to calculate the total sales by Region Names. Finally we drop the table Regions, as in this .qvw file we only need the table named **Total** for data analysis.

The screenshot shows the QlikView script editor window titled "Edit Script [QV1]". The main pane displays the following QlikView script:

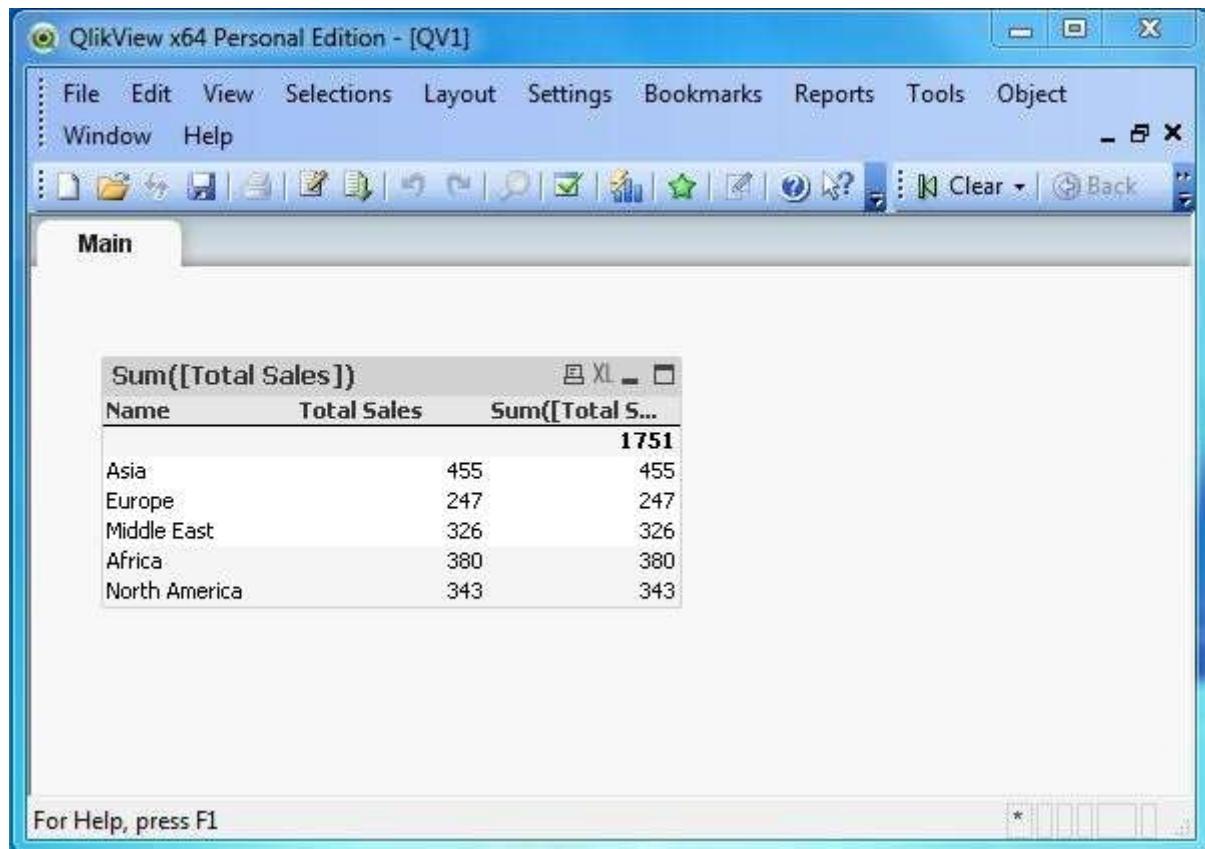
```
Regions:
LOAD * INLINE [
    Name, Sales, Year
    Asia, 250, 2014
    Europe, 138, 2014
    Middle East, 211, 2011
    Africa, 79, 2012
    North America, 321, 2011
    Middle East, 115, 2014
    Asia, 205, 2011
    Europe, 109, 2012
    Africa, 301, 2014
    North America, 22, 2011
];
Total:
Load
    Name, SUM(Sales) as [Total Sales]
Resident Regions Group By Name;

Drop Table Regions;
```

Below the script editor is a "Data" tab of a configuration dialog. The "Database" section is set to "ODBC" with "Force 32 Bit" checked. The "Data from Files" section contains buttons for "Table Files...", "QlikView File...", "Web Files...", and "Field Data...". At the bottom are "OK", "Cancel", and "Help" buttons.

## Table Box Data

On creating a Table Box Sheet Object, we see the data that is read from the resident data load option.



The screenshot shows the QlikView x64 Personal Edition interface with the title bar "QlikView x64 Personal Edition - [QV1]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. The toolbar below has various icons for file operations like Open, Save, Print, and search. The main window is titled "Main" and contains a table box. The table has a header row with columns "Name", "Total Sales", and "Sum([Total S...])". The data rows show sales for Asia, Europe, Middle East, Africa, and North America. The total value "1751" is displayed in the last column of the header row. The table is styled with alternating row colors.

Name	Total Sales	Sum([Total S...])
Asia	455	455
Europe	247	247
Middle East	326	326
Africa	380	380
North America	343	343

## 12. QlikView – Preceding Load

QlikView Preceding load is a load type in which we use a load statement, which takes the columns of another load statement present in the same script. The data read by the first Load statement, which is at the bottom of the script editor window and then used by the load statements above it.

### Load Script

The below given screen shot shows the script for data, which is loaded as **Inline data** and then the max function is applied to one of the columns. The load statement at the bottom makes the data available in QlikView's memory, which is used by the second load statement above the first load statement. The second load statement applies the max function with group by clause.

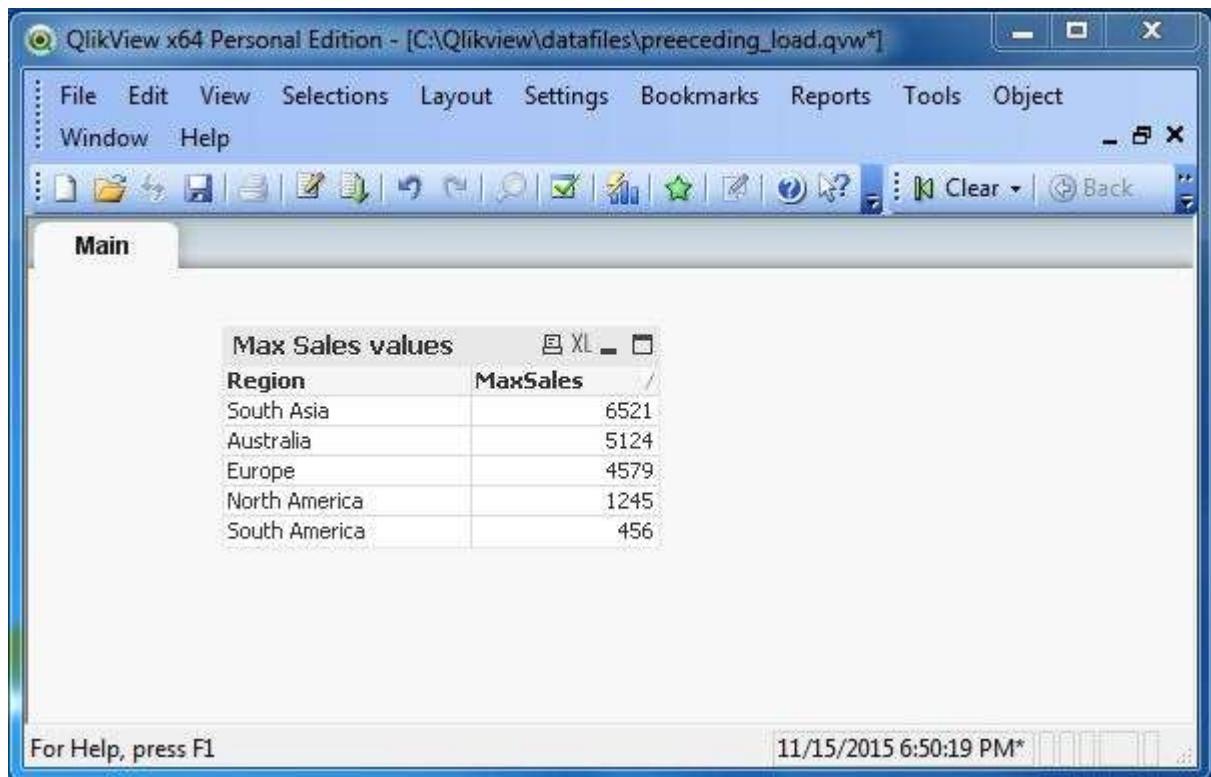
The screenshot shows the 'Edit Script' window in QlikView. The title bar reads 'Edit Script [C:\Qlikview\datafiles\preceeding\_load.qvw]'. The menu bar includes File, Edit, Insert, Tab, Tools, and Help. The toolbar contains icons for Reload, Debug, and various file operations. The main area is titled 'Main' and contains the following QlikView script:

```
1 MaxRegions:  
2   LOAD Region,max(SaleAmount) as MaxSales  
3   Group by Region;  
4  
5 AllRegions:  
6   LOAD * INLINE [  
7     ProductID,Region,SaleAmount  
8     1,Europe,4579  
9     2,Europe,4125  
10    3,South Asia,6521  
11    4,South Asia,4125  
12    5,Australia,5124  
13    6,North America,1245  
14    7,South America,456  
15    8,North America,1245  
16  ];  
17 ];
```

Below the script, there is a 'Data' configuration panel with tabs for Data, Functions, Variables, and Settings. The Data tab shows settings for a database connection via ODBC, with options for Force 32 Bit, Connect, Select, and Data from Files (Relative Paths, Use FTP, Table Files, QlikView File, Web Files, Field Data). At the bottom are OK, Cancel, and Help buttons.

## Table Box Data

On creating a Table Box Sheet Object, we see the data that is read from the Inline data load option.



The screenshot shows the QlikView x64 Personal Edition interface. The main window title is "QlikView x64 Personal Edition - [C:\Qlikview\datafiles\preceding\_load.qvw\*]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. The toolbar below the menu contains various icons for file operations like Open, Save, Print, and zoom. The main workspace is titled "Main" and contains a table titled "Max Sales values". The table has two columns: "Region" and "MaxSales". The data rows are:

Region	MaxSales
South Asia	6521
Australia	5124
Europe	4579
North America	1245
South America	456

At the bottom left of the workspace, it says "For Help, press F1". At the bottom right, the date and time are displayed as "11/15/2015 6:50:19 PM\*".

## 13. QlikView – Incremental Load

As the volume of data in the data source of a QlikView document increases, the time taken to load the file also increases which slows down the process of analysis. One approach to minimize this time taken to load data is to load only the records that are new in the source or the updated ones. This concept of loading only the new or changed records from the source into the QlikView document is called **Incremental Load**.

To identify the new records from source, we use either a sequential unique key or a date time stamp for each row. These values of unique key or data time field has to flow from the source file to QlikView document.

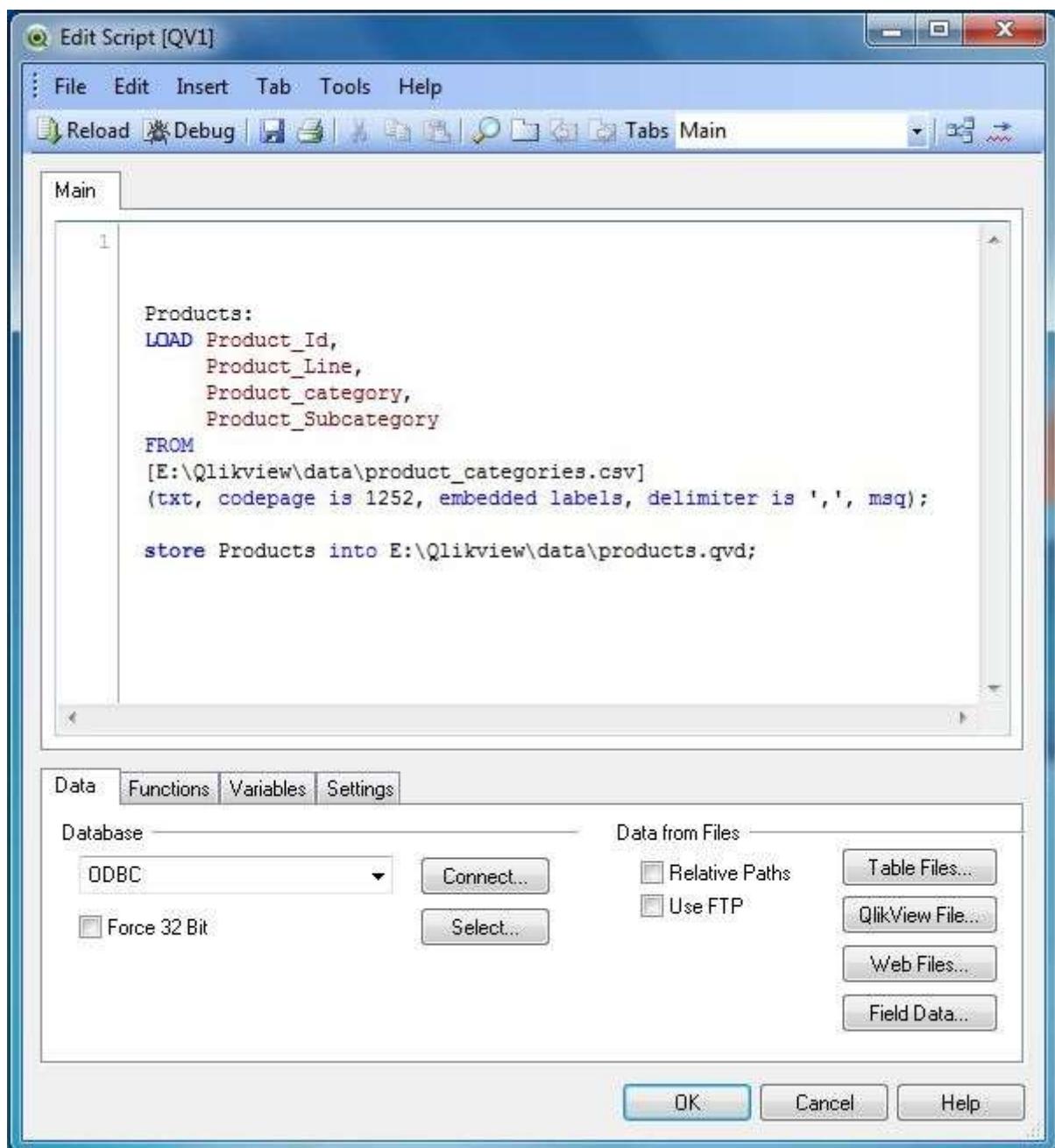
Let us consider the following source file containing product details in a retail store. Save this as a .csv file in the local system where it is accessible by QlikView. Over a period of time some more products are added and the description of some product changes.

```
Product_Id,Product_Line,Product_category,Product_Subcategory
1,Sporting Goods,Outdoor Recreation,Winter Sports & Activities
2,"Food, Beverages & Tobacco",Food Items,Fruits & Vegetables
3,Apparel & Accessories,Clothing,Uniforms
4,Sporting Goods,Athletics,Rugby
5,Health & Beauty,Personal Care
6,Arts & Entertainment,Hobbies & Creative Arts,Musical Instruments
7,Arts & Entertainment,Hobbies & Creative Arts,Orchestra Accessories
8,Arts & Entertainment,Hobbies & Creative Arts,Crafting Materials
9,Hardware,Tool Accessories,Power Tool Batteries
10,Home & Garden,Bathroom Accessories,Bath Caddies
11,"Food, Beverages & Tobacco",Food Items,Frozen Vegetables
12,Home & Garden,Lawn & Garden,Power Equipment
```

### Loading the Data into QlikView

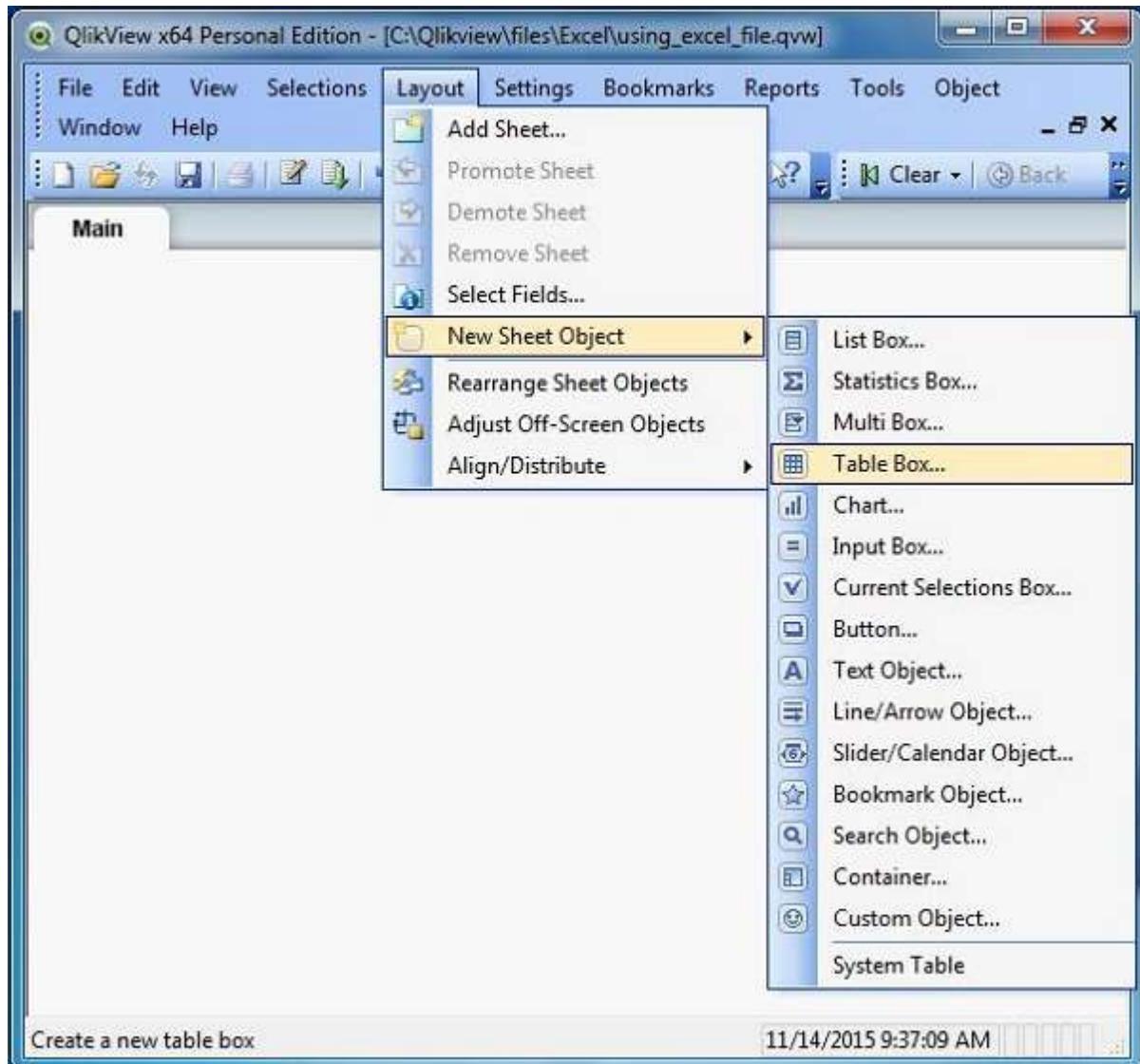
64

We will load the above CSV file using the script editor (Control+E) by choosing the Table Files option as shown below. Here we also save the data into a QVD file in the local system. Save the QlikView document as a .qvw file.



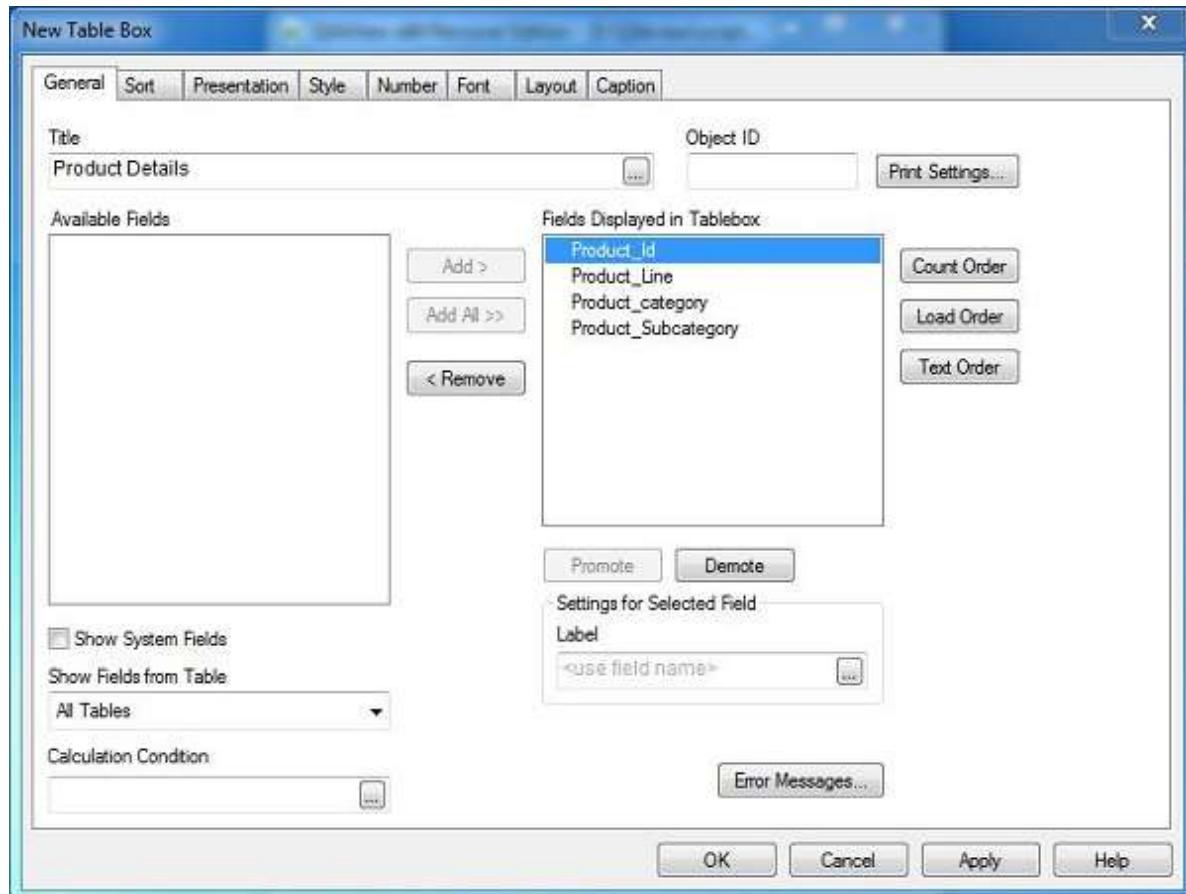
## Verifying the Data Loaded.

We can check the data loaded to QlikView document by creating a sheet object called **Table Box**. This is available in the Layout menu and **New Sheet Objects** submenu.



## Creating the Table Layout

On selecting the **Table Box** sheet object, we get to the next screen, which is used to select the columns and their positions in the table to be created. We choose the following columns and their positions and click Finish.



## Viewing the Existing Data

The following chart showing the data as laid out in the previous step appears.

The screenshot shows the QlikView interface with a title bar 'QlikView x64 Personal Edition - [QV1]'. The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. Below the menu is a toolbar with various icons. The main area is titled 'Main' and contains a table titled 'Product Details' with the following data:

Product_Id	Product_Line	Product_category	Product_Subcategory
1	Sporting Goods	Outdoor Recreation	Winter Sports & Activities
2	Food, Beverages & Tobacco	Food Items	Fruits & Vegetables
3	Apparel & Accessories	Clothing	Uniforms
4	Sporting Goods	Athletics	Rugby
5	Health & Beauty	Personal Care	
6	Arts & Entertainment	Hobbies & Creative Arts	Musical Instruments
7	Arts & Entertainment	Hobbies & Creative Arts	Orchestra Accessories
8	Arts & Entertainment	Hobbies & Creative Arts	Crafting Materials
9	Hardware	Tool Accessories	Power Tool Batteries
10	Home & Garden	Bathroom Accessories	Bath Caddies
11	Food, Beverages & Tobacco	Food Items	Frozen Vegetables
12	Home & Garden	Lawn & Garden	Power Equipment

At the bottom left of the main area, it says 'For Help, press F1'.

## Updating the Source Data

Let us add the following three more records to the source data. Here, the Product IDs are the unique numbers, which represent new records.

```
13,Office Supplies,Presentation Supplies,Display
14,Hardware,Tool Accessories,Jigs
15,Baby & Toddler,Diapering,Baby Wipes
```

## Incremental Load Script

Now, we write the script to pull only the new records from the source.

```
// Load the data from the stored qvd.
Stored_Products:
LOAD Product_Id,
    Product_Line,
    Product_category,
    Product_Subcategory
FROM
[E:\Qlikview\data\products.qvd]
```

```
(qvd);
//Select the maximum value of Product ID.

Max_Product_ID:
Load max(Product_Id) as MaxId
resident Stored_Products;
//Store the Maximum value of product Id in a variable.

Let MaxId = peek('MaxId',-1);
drop table Stored_Products;
//Pull the rows that are new.

NewProducts:
LOAD Product_Id,Product_Line, Product_category,Product_Subcategory
from [E:\Qlikview\data\product_categories.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq)
where Product_Id > $(MaxId);
//Concatenate the new values with existing qvd.

Concatenate
LOAD Product_Id,Product_Line, Product_category,
Product_Subcategory
FROM [E:\Qlikview\data\products.qvd](qvd);
//Store the values in qvd.

store NewProducts into [E:\Qlikview\data\products.qvd](qvd);
```

The above script fetches only the new records, which are loaded and stored into the qvd file. As we see the records with the new Product IDs 13, 14 and 15.

QlikView x64 Personal Edition - [QV1]

Main

Product\_Id Product\_Line Product\_Category Product\_Subcategory

Product_Id	Product_Line	Product_Category	Product_Subcategory
1	Sporting Goods	Outdoor Recreation	Winter Sports & Activities
2	Food, Beverages & Tobacco	Food Items	Fruits & Vegetables
3	Apparel & Accessories	Clothing	Uniforms
4	Sporting Goods	Athletics	Rugby
5	Health & Beauty	Personal Care	
6	Arts & Entertainment	Hobbies & Creative Arts	Musical Instruments
7	Arts & Entertainment	Hobbies & Creative Arts	Orchestra Accessories
8	Arts & Entertainment	Hobbies & Creative Arts	Crafting Materials
9	Hardware	Tool Accessories	Power Tool Batteries
10	Home & Garden	Bathroom Accessories	Bath Caddies
11	Food, Beverages & Tobacco	Food Items	Frozen Vegetables
12	Home & Garden	Lawn & Garden	Power Equipment
13	Office Supplies	Presentation Supplies	Display
14	Hardware	Tool Accessories	Jigs
15	Baby & Toddler	Diapering	Baby Wipes

For Help, press F1

## 14. QlikView – Data files (QVD)

One of the important features of QlikView, which makes it so distinguished is the ability to store very large amount of data in a very compressed size and store it along with the QlikView documents. Therefore, once the document is created we need not connect to the data source, as the data is already stored along with the layout of the document. This is achieved through QVD file, which is a flat file stored with the .qvd extension. A QVD file stores data for one QlikView document and it is created using the script editor available in the QlikView document.

### **Advantages of Using QVD Files**

---

The advantages of using QVD files in QlikView are as follows:

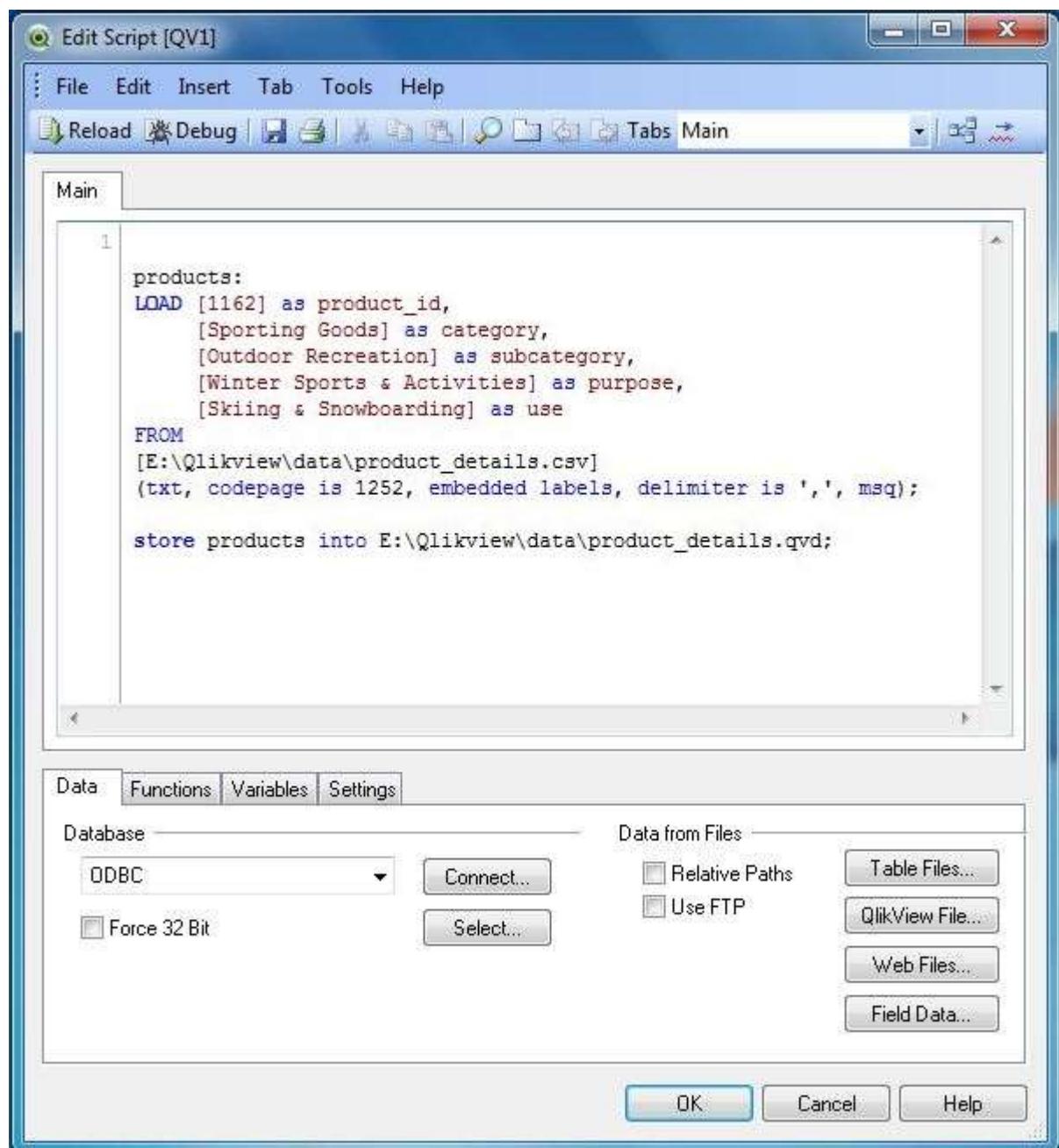
- Faster Loading of Data
- Gracefully support scaling up as the data volume grows
- Used in incremental load
- Data from multiple sources can be clubbed to one data set
- Extract data in parallel

### **Creating QVD Files**

---

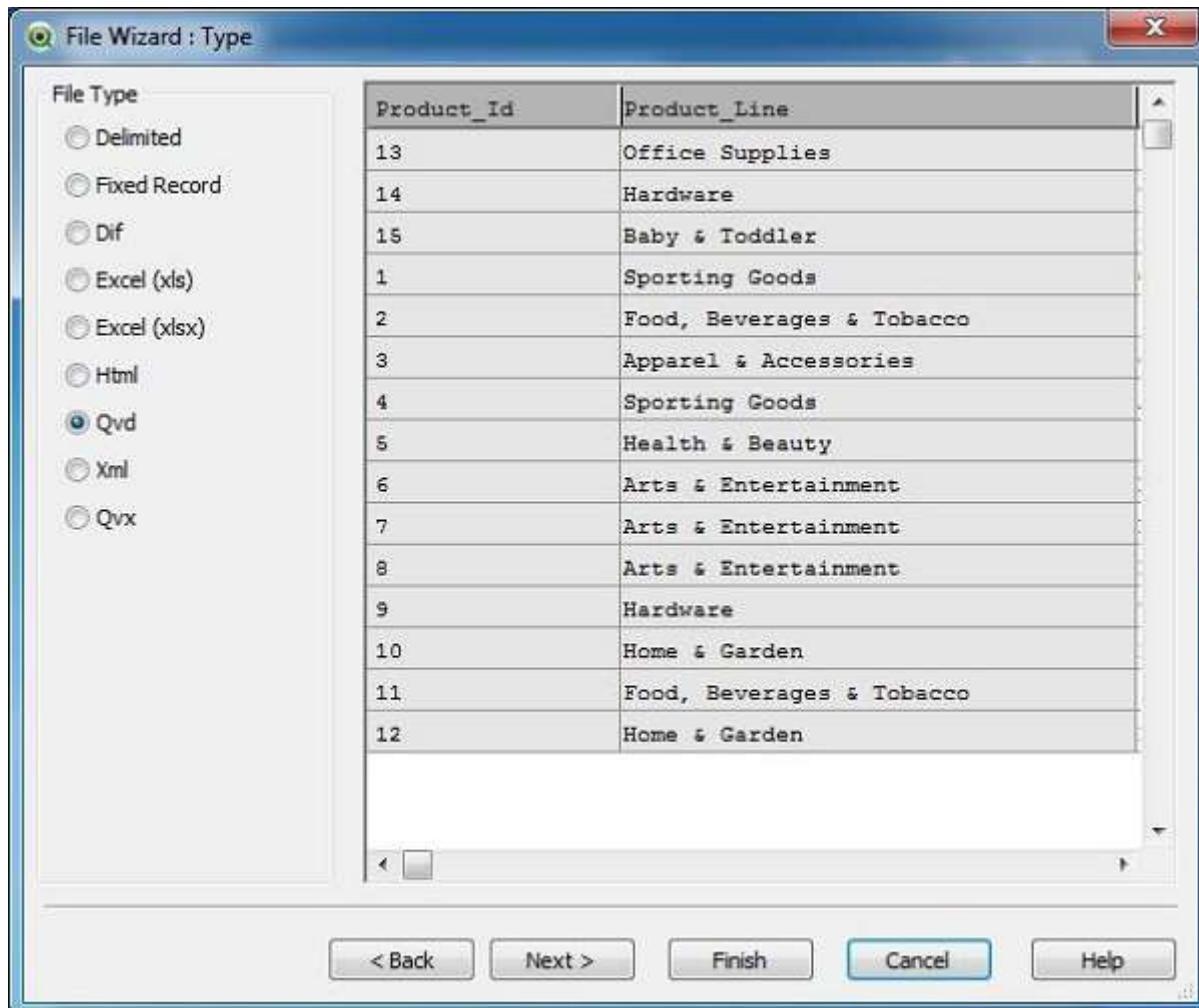
QVD files are created using the STORE statement during the loading of QlikView files. This statement creates a single qvd file, which gets stored in the specified location as a file; separate than the QVW file through which it is created.

Given below is an example of storing the qvd file after the data is loaded into the QlikView document by reading a source file.



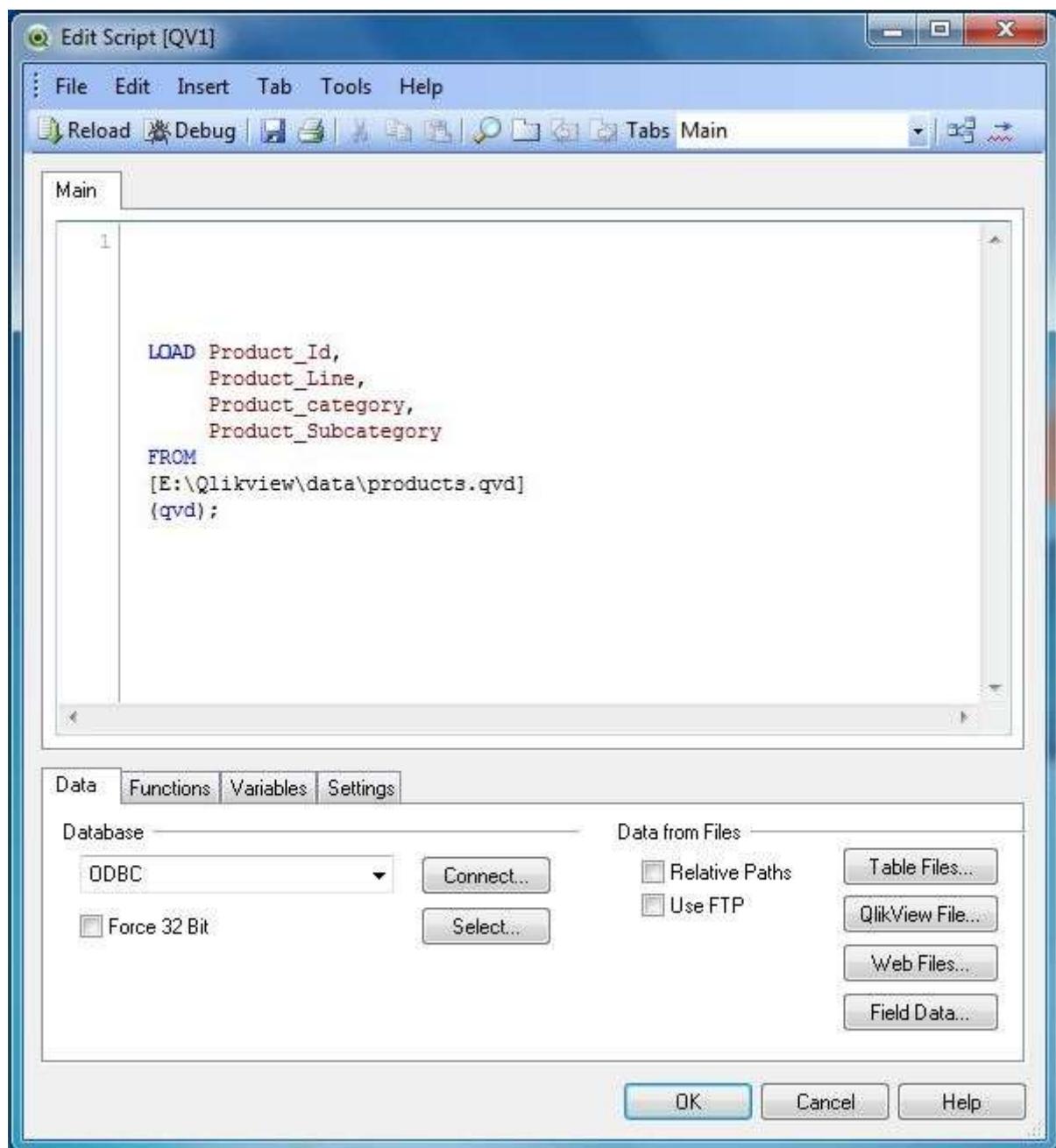
## Using QVD Files

A QVD file is loaded to a QlikView document in a similar way as other files like CSV, Excel and delimited files are used. We use the the **Open** option available under the **File** menu and browse for the QVD file we created before. On opening it gives us a window to see the data, select the column headers and do any data transformation required.



## QVD File Loader Script

On clicking Finish, the edit script window appears which shows the code used to load the QVD file. We can edit this code further. For example, to get only the few of the columns to be displayed or apply any inbuilt function etc. Click finish to load the file into the current QlikView document. Save the QlikView document as **use\_qvd.qvw**.



## Displaying Data from QVD File

Reload the document by using **Control+R** and choose the menu option **Layout -> New Sheet Objects -> Table Box**. A window appears showing all the columns from the table present in the QVD file. Select "**Add All**" to add all the columns to the display table. Use the "**Promote/Demote**" option to change the order of the columns. Click "Finish". The following screen appears showing the content of the QVD file.

QlikView x64 Personal Edition - [QV1]

Main

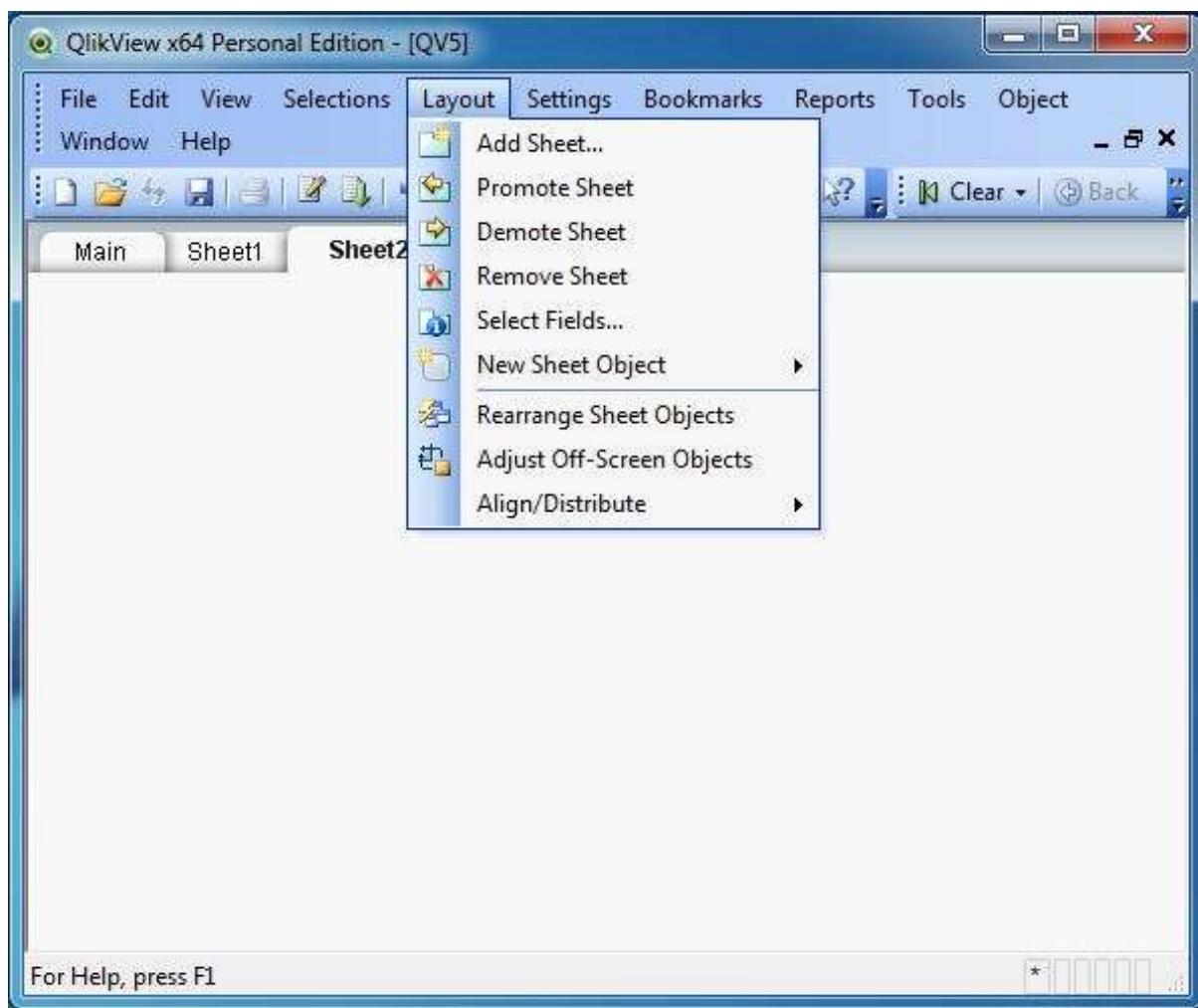
Product_Id	Product_Line	Product_Category	Product_Subcategory
1	Sporting Goods	Outdoor Recreation	Winter Sports & Activities
2	Food, Beverages & Tobacco	Food Items	Fruits & Vegetables
3	Apparel & Accessories	Clothing	Uniforms
4	Sporting Goods	Athletics	Rugby
5	Health & Beauty	Personal Care	
6	Arts & Entertainment	Hobbies & Creative Arts	Musical Instruments
7	Arts & Entertainment	Hobbies & Creative Arts	Orchestra Accessories
8	Arts & Entertainment	Hobbies & Creative Arts	Crafting Materials
9	Hardware	Tool Accessories	Power Tool Batteries
10	Home & Garden	Bathroom Accessories	Bath Caddies
11	Food, Beverages & Tobacco	Food Items	Frozen Vegetables
12	Home & Garden	Lawn & Garden	Power Equipment
13	Office Supplies	Presentation Supplies	Display
14	Hardware	Tool Accessories	Jigs
15	Baby & Toddler	Diapering	Baby Wipes

For Help, press F1

# QlikView Report Interface

## 15. QlikView – Sheet and Objects

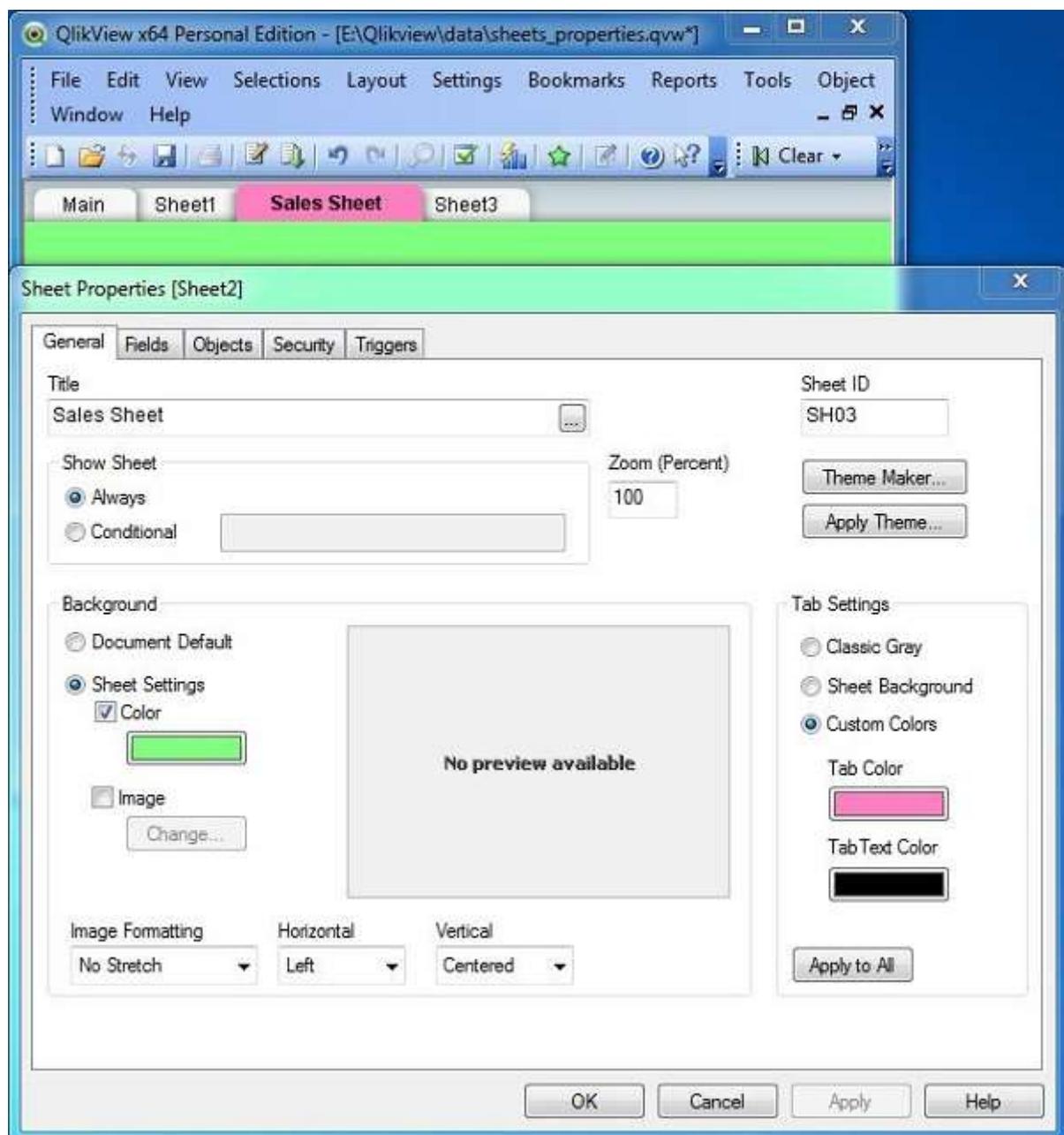
Every QlikView document is made of at least one worksheet called **Main**. We can add more sheets, which are like many pages of the same QlikView document. Sheets help us display multiple data formats like - multiple charts or multiple tables. Each sheet can contain various sheet objects. In addition, sheets can be rearranged using **Promote Sheet/Demote Sheet** option and can be removed from the QlikView document using **Remove Sheet** option.



### Sheet Properties

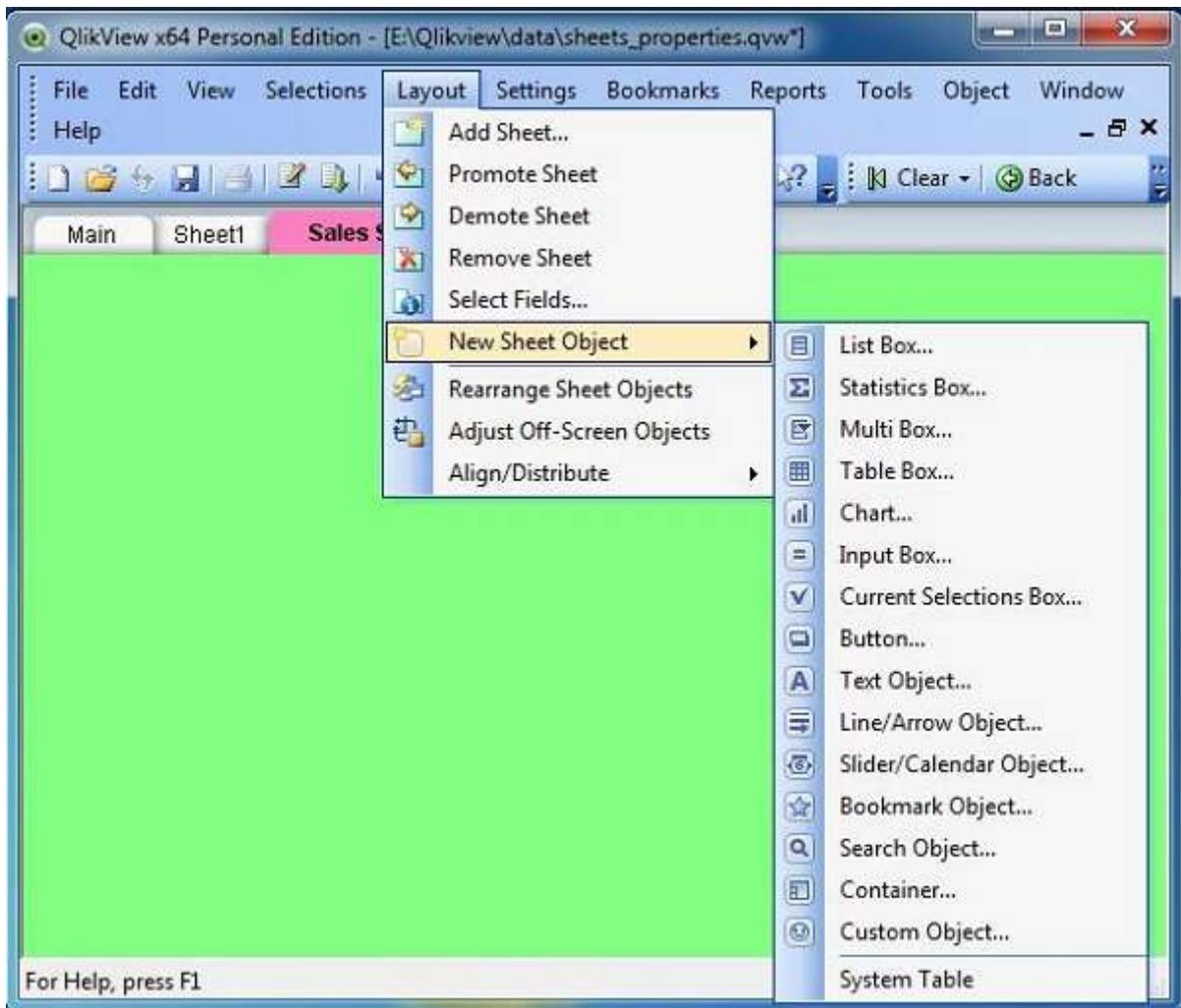
Sheets have various properties, which can be set to customize the sheets. For example, we can set the name of the sheets and its colors. Right click anywhere in the sheet and choose the Properties option. Then choose the following properties.

- **Sheet Settings -> Color** - This will set the background color of the Sheet.
- **Tab Settings -> Custom Colors** - This will set the color for the Tab where the Sheet name appears.
- **Title** - This will set the name of the Sheet.



## Creating Sheet Objects

Sheet Objects are the QlikView data elements that are embedded in the sheet. They display the data that is loaded into the QlikView's memory. Each sheet object is tied to a data source and one or more of its columns. Sheet Objects are created from the **layout** menu as shown below.

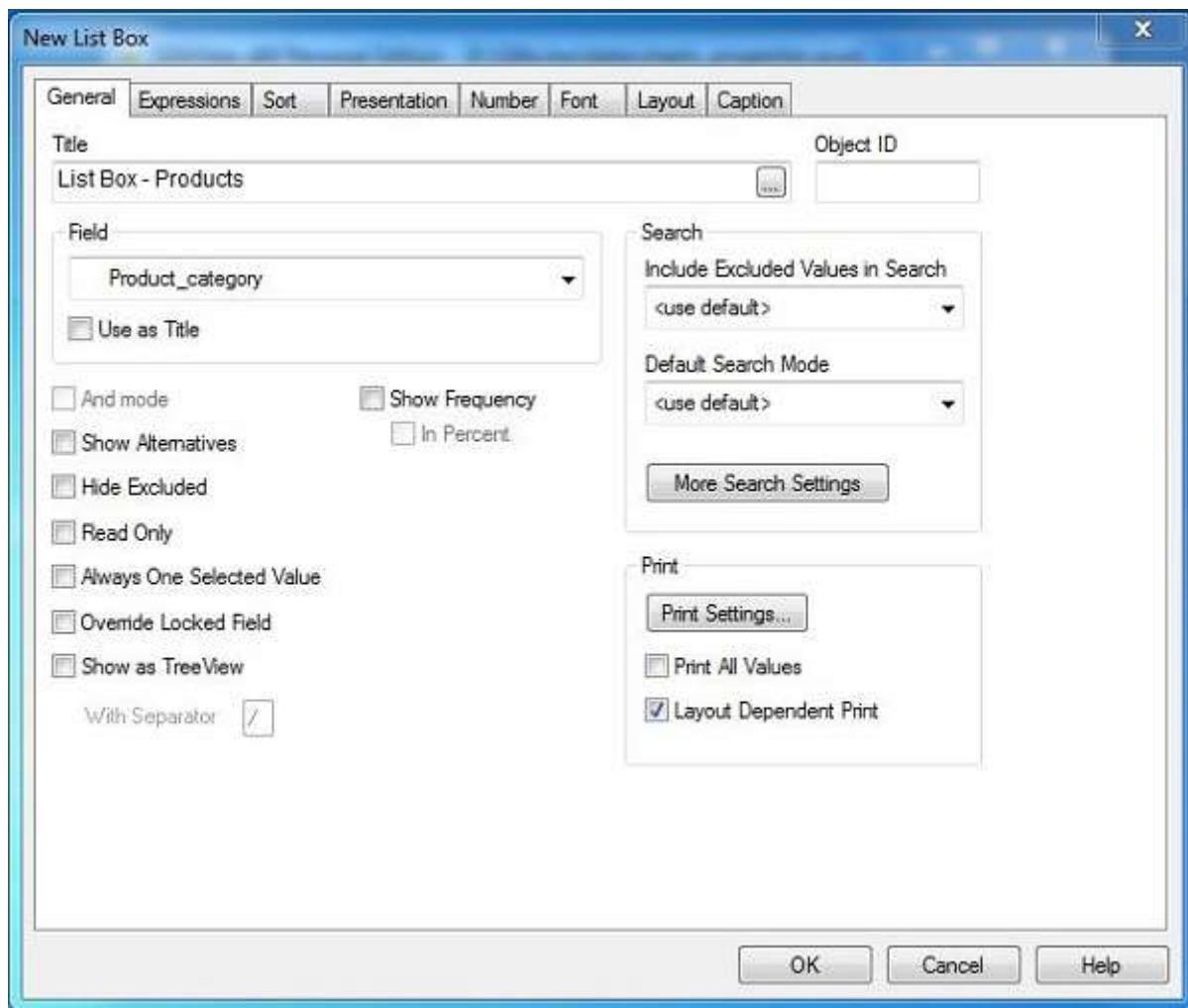


## Using Sheet Objects

Sheet Objects display the data from a data source and all the objects in a sheet are associated with each other. Let us create a List Box and a Multi Box and see this association on action.

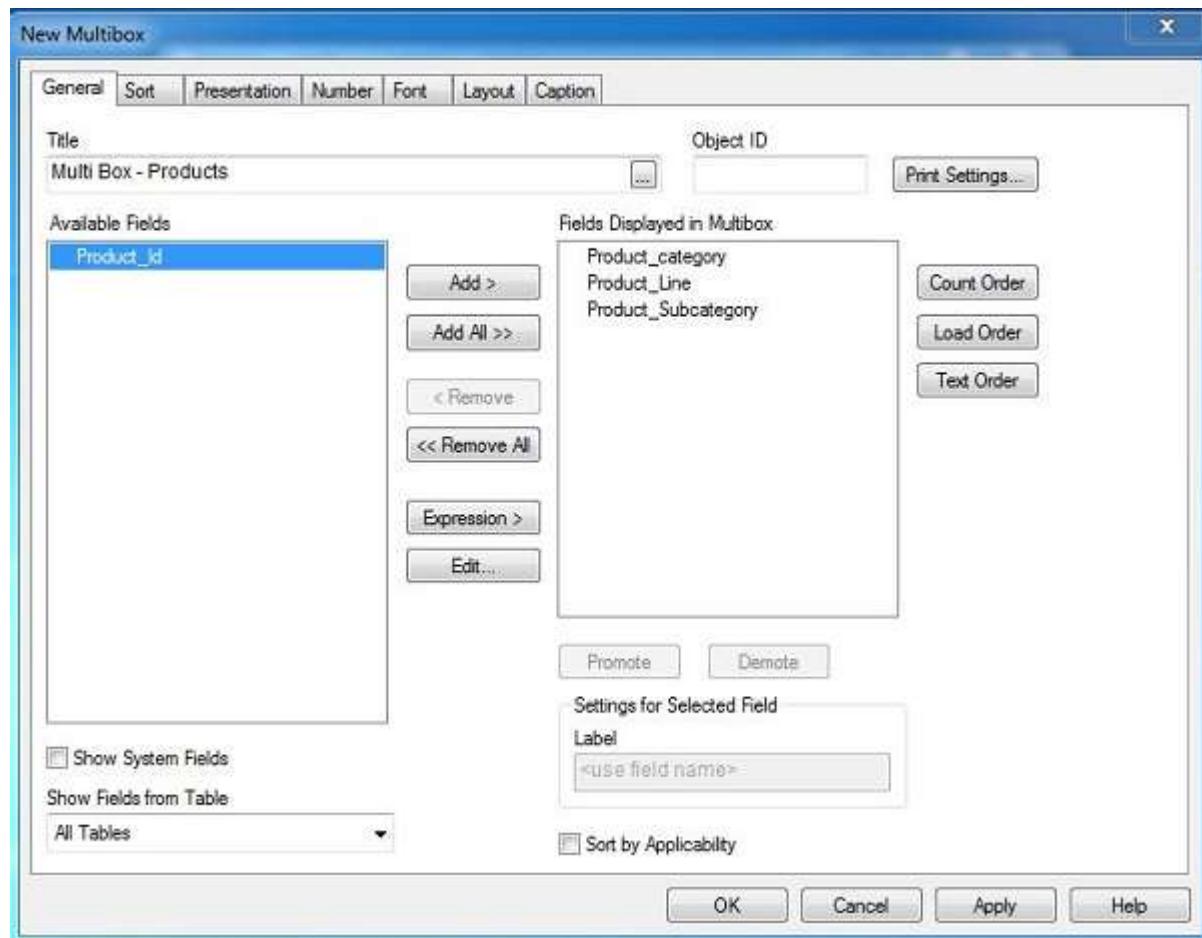
### Creating List Box

The List box displays data from a column of a table available in QlikView memory. Choose the option **List Box** from the Add Sheet Objects option and set the properties as given below.

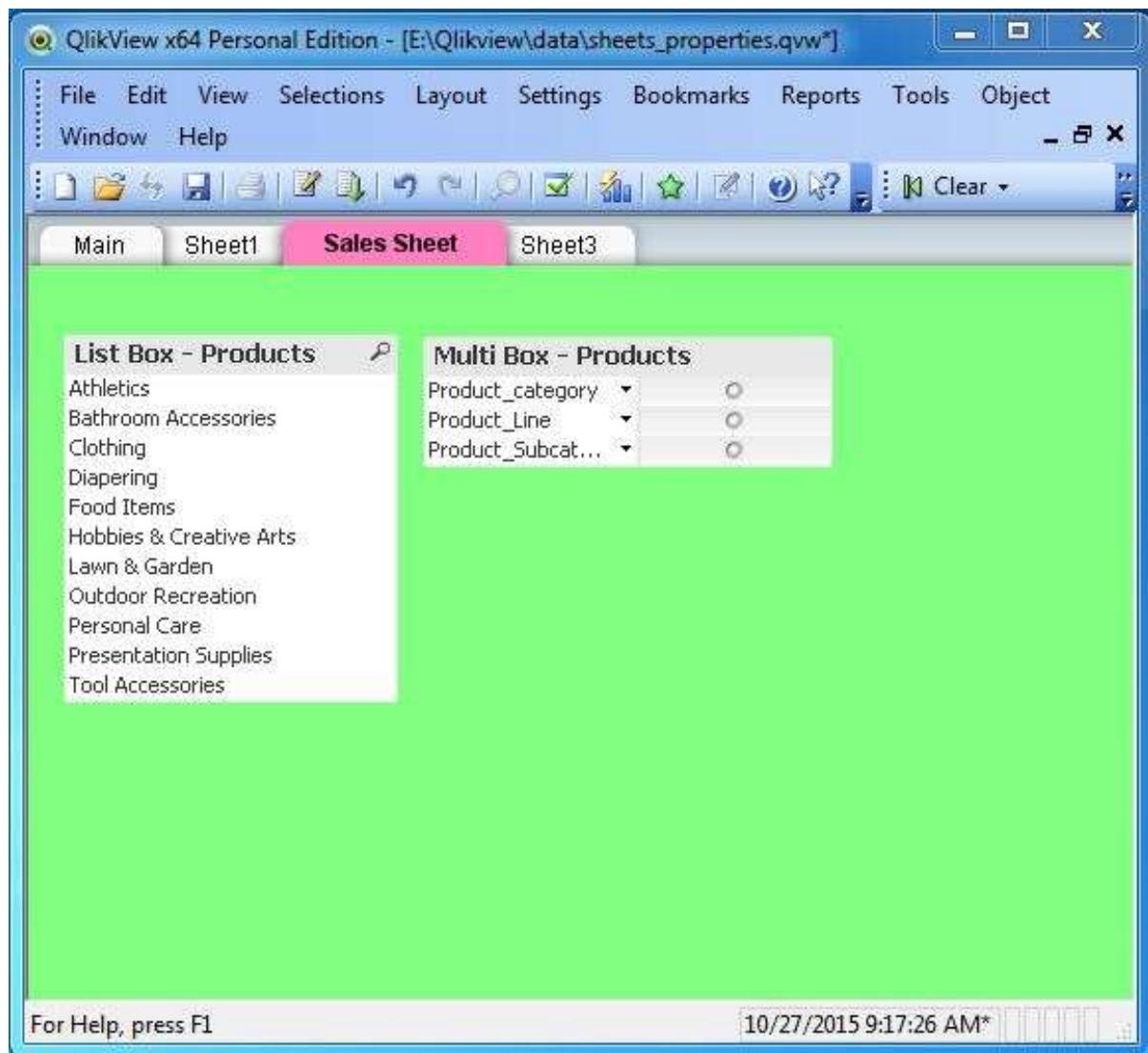


## Creating Multi Box

A Multi Box represents data from multiple columns from a table. Choose the option **Multi Box** from the Add Sheet Objects option and set the properties as shown below.

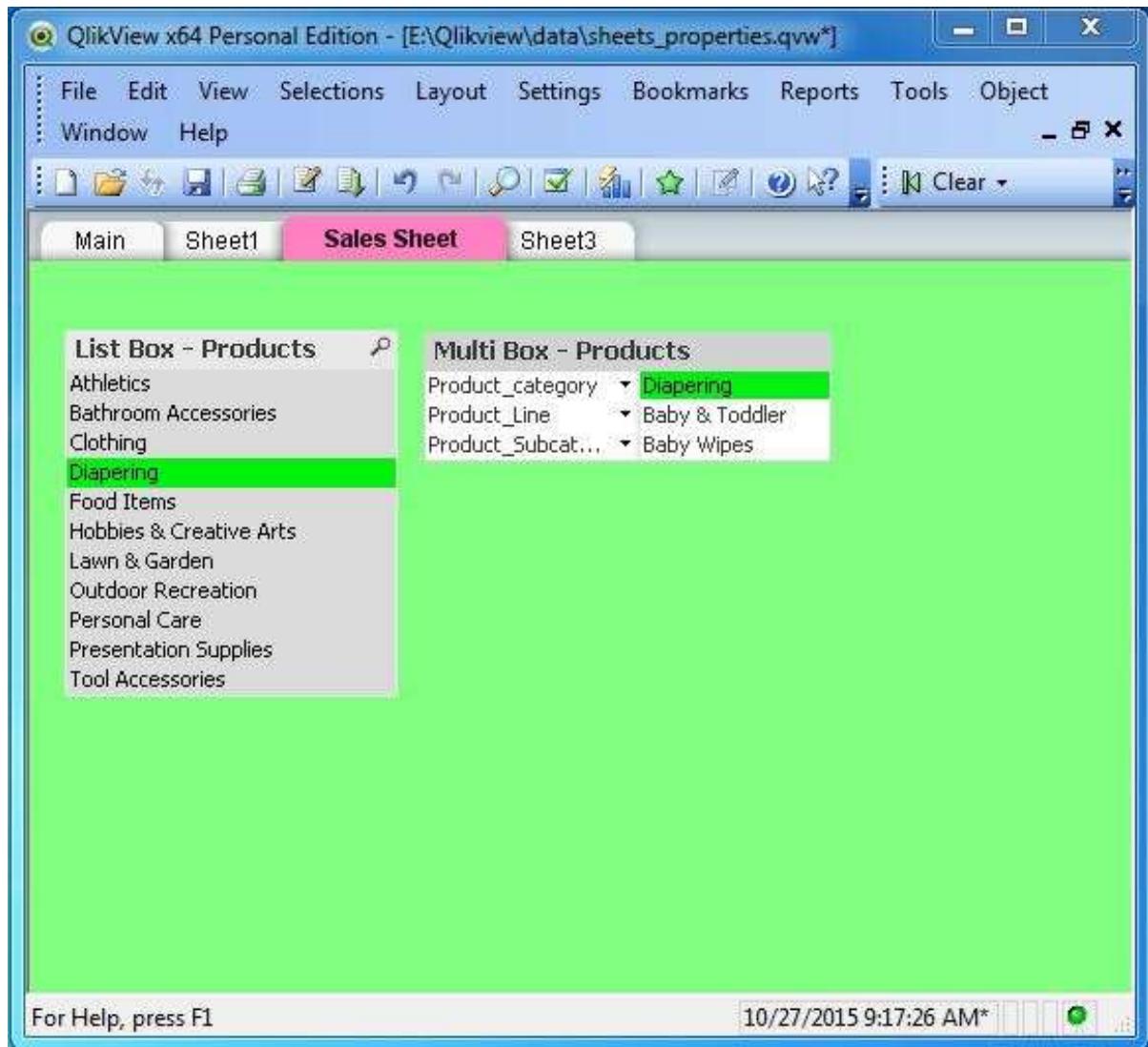


On completing the above given steps, the following window appears which shows both the sheet objects.



## Association between Sheet Objects

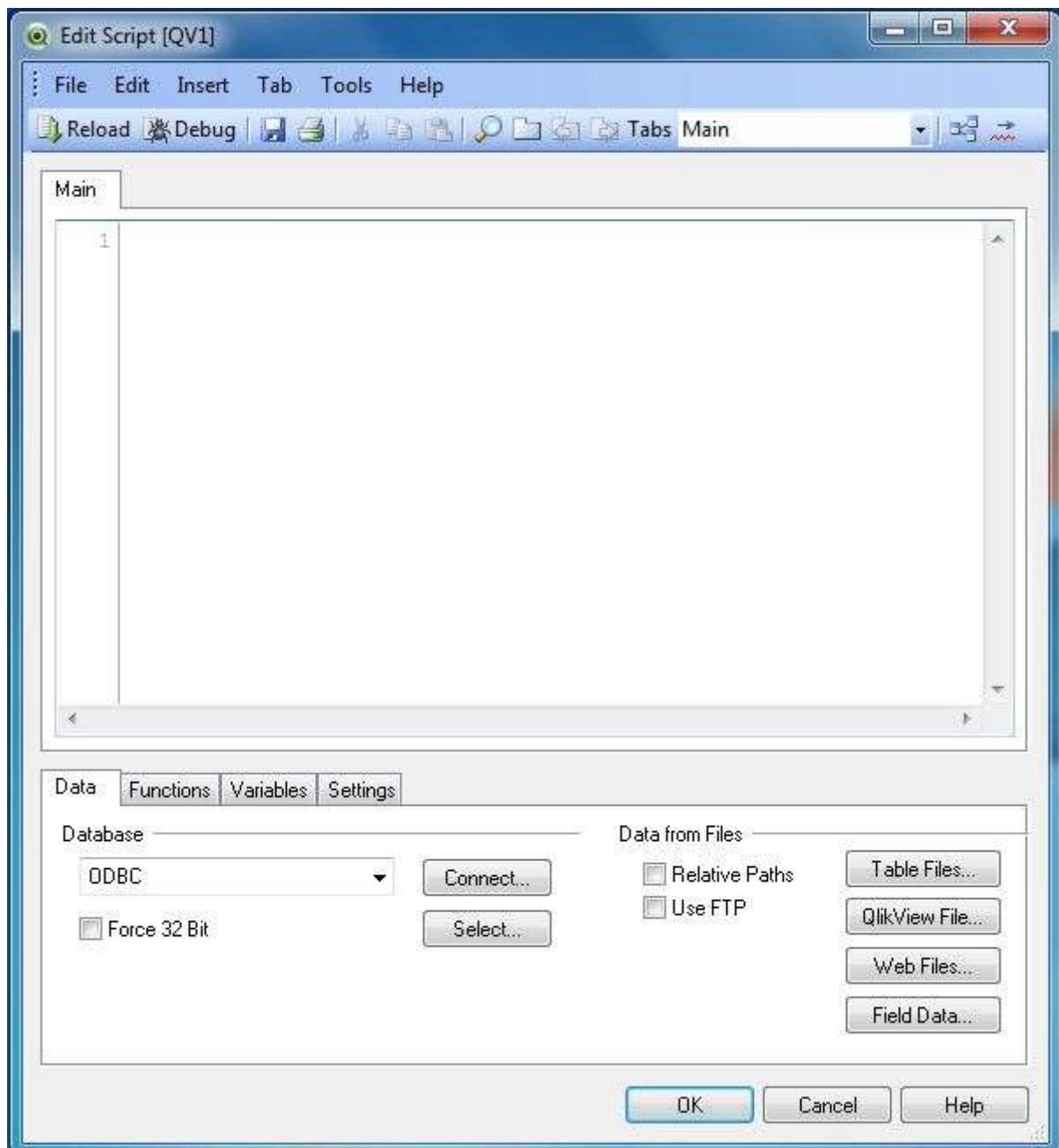
We can see how the sheet objects are linked to each other by choosing the one option from the Multi Box, which highlights the associated row in the List Box. Let us choose "Diapering" under the Product Category drop down list in Multi Box. The window shown below appears.



## 16. QlikView – Scripting

Scripting is a very powerful feature in QlikView, which enables the control of the data load options and data transformations. It enables the use of many inbuilt functions available in QlikView and creates subroutines to be used across multiple scripts within a QlikView document.

Scripting is done using the Script Editor. It is accessed from the File menu using **Script Editor**. We can also press **Control + E** to open the script editor window. It prepopulates some data formats that are set as default formats for the data to be processed. For example, the Thousand separator is a comma and date is in Month-day-year format. These can be changed to suit the incoming data as per the need.



## Script Editor Features

---

Script editor has many features, which are accessed from the menu in the script editor window, which is a different menu from the main menu. Given below is a list of important features.

### File Menu

- **Reload** - Reloads the script and fetches the new data.
- **Upper/Lower Case** - Converts the case of words as QlikView is case sensitive.
- **Comment** - Used to comment blocks of code.

### Edit Menu

- **Clear Entire Script** - Clears the active script tab.
- **Open Script File** - Opens the saved script files.
- **Table Viewer** – Used to see the tabular view of the data being loaded.

### Insert Menu

- **Environment Variables** - Inserts a standard list of Environment variables.
- **Script Files** - Allows to browse for script files and insert them.
- **Connect/Disconnect Statement** - Used to connect or disconnect from external databases.

### Tab Menu

- **Insert Tab** - Inserts a Tab at the current cursor position.
- **Promote/Demote Tab** - Allows to move the tabs from left to right and vice versa.
- **Merge with Previous** - Used to merge the content of active tag with previous tab.

### Tools Menu

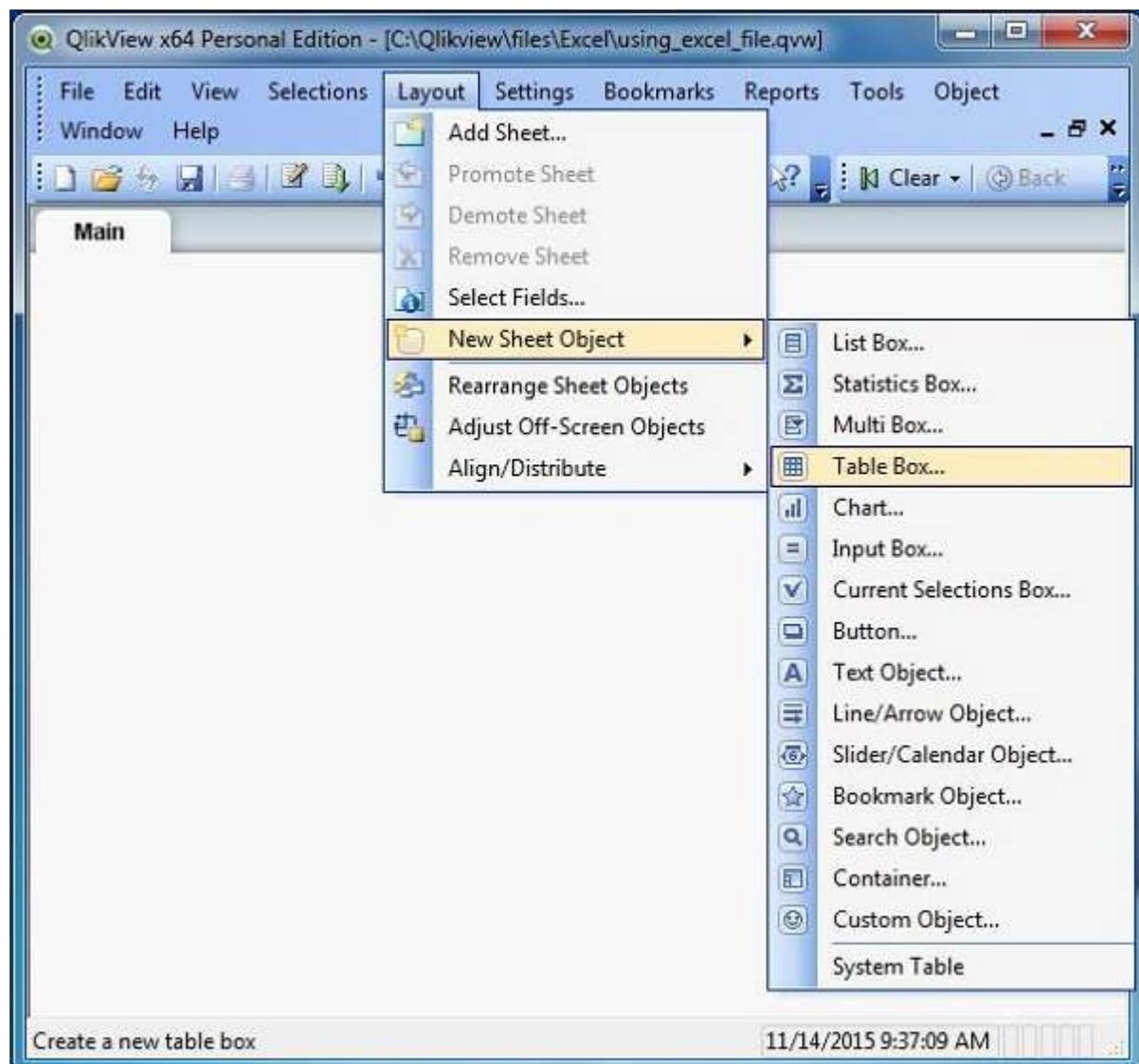
- **ODBC Administrator 64 bit/ODBC Administrator 32 bit** - Allows to set the correct DSN information for data sources.
- **Editor Preferences** - Allows you to configure the text font and size, help features, shortcuts, default-scripting engine etc.
- **Syntax Check** - Used to validate the syntax of the script code.

# 17. QlikView – Functions

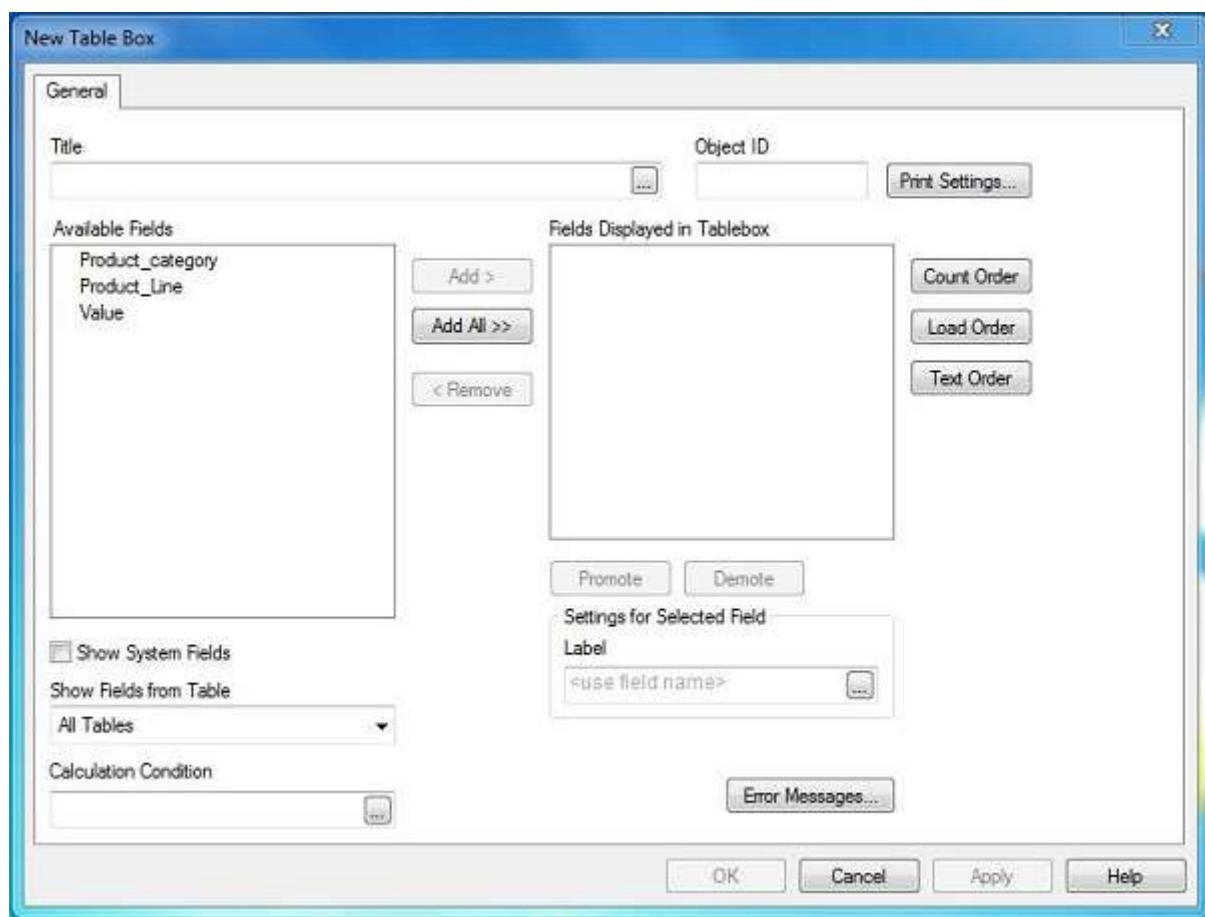
QlikView has many built-in functions, which are available to be applied to data that is already available in memory. These functions are organized into many categories and the syntax of the function appears as soon as it is selected. We can click on the Paste button to get the expression into the editor and supply the arguments.

## Create Table Box

Create a Table Box by following the menu as shown in the screen shot given below.

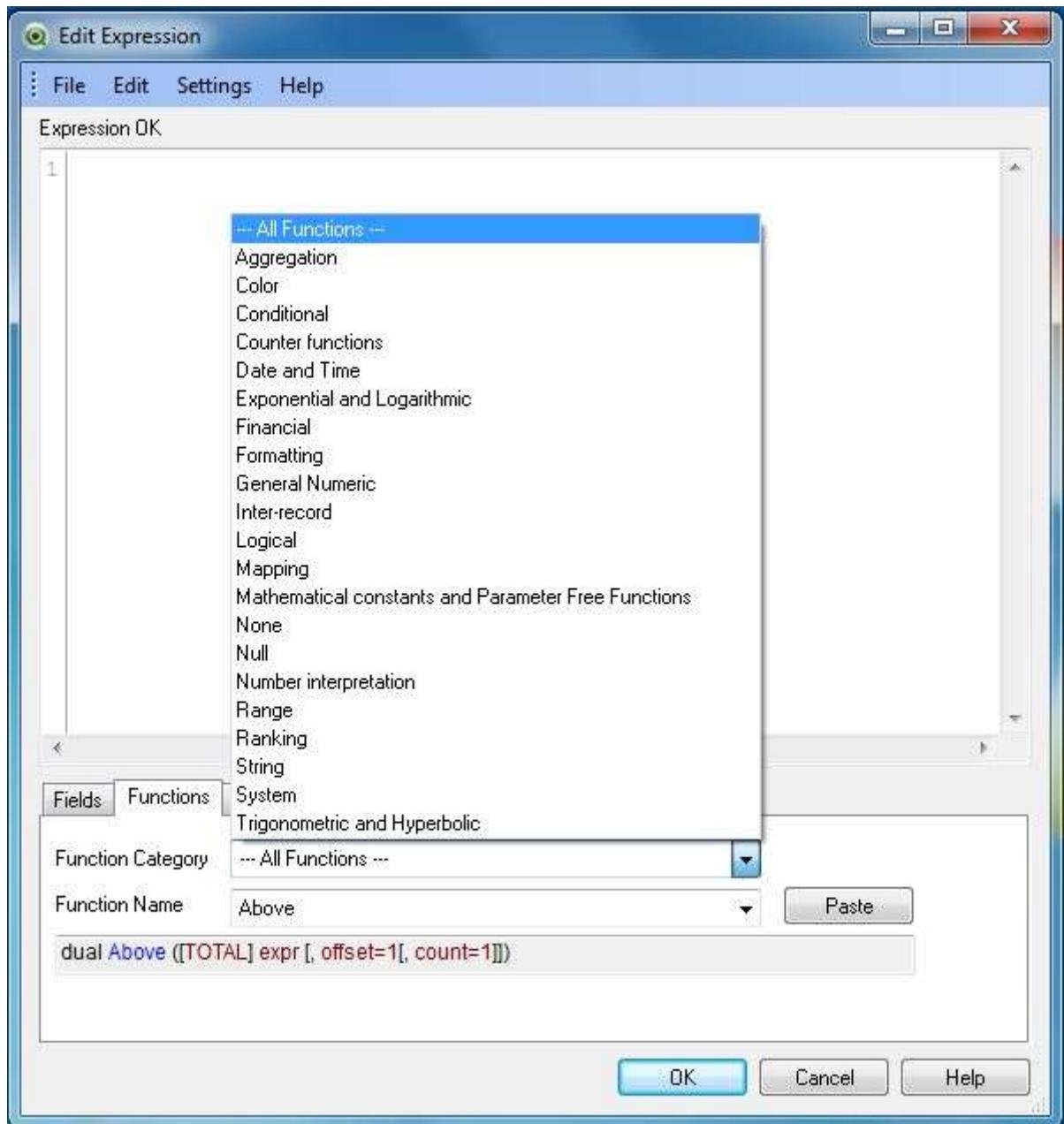


On completing the above given step, we get a window to show the Calculation condition at the bottom left.

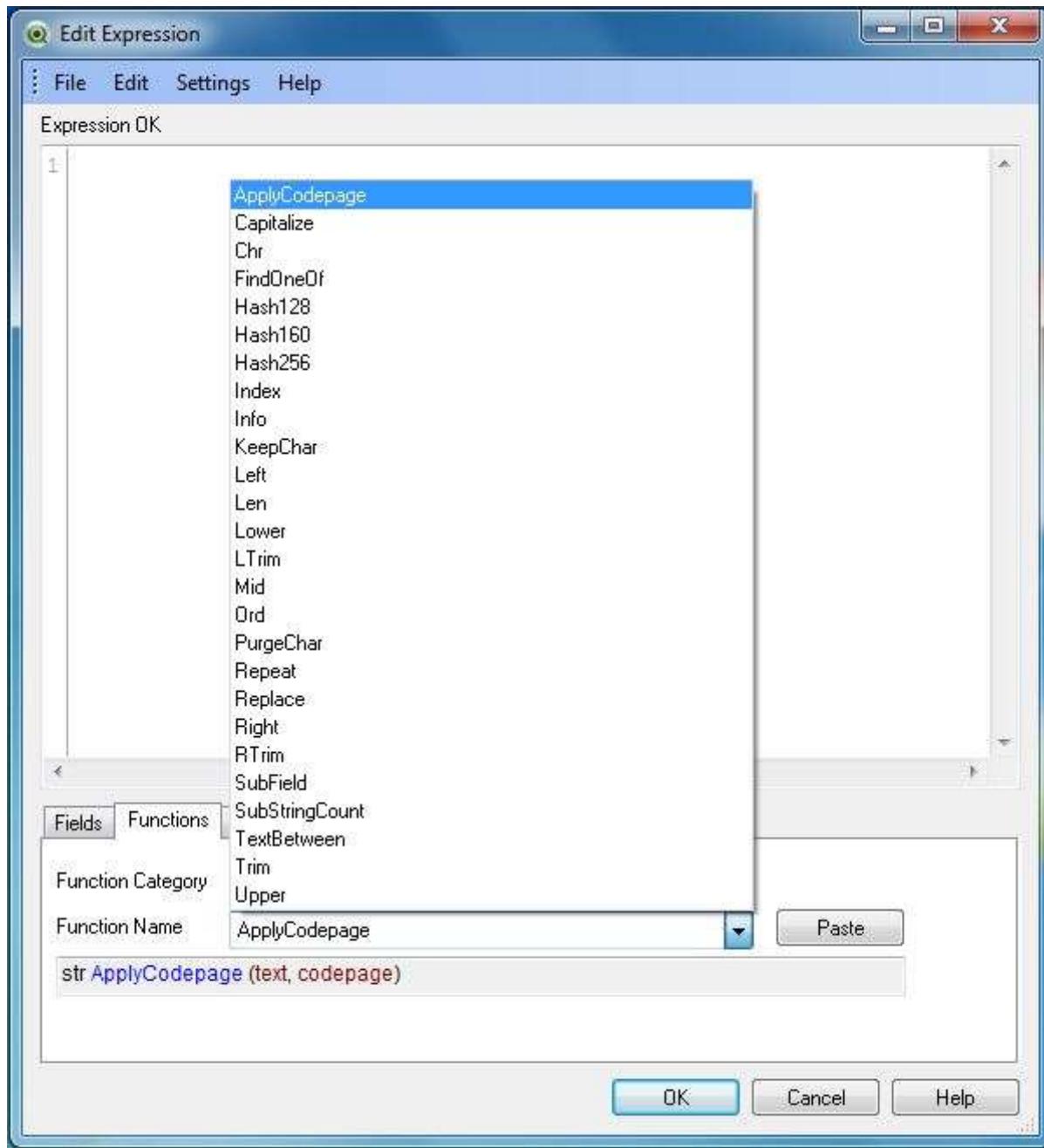


## List of Functions

Click on the button next to calculation condition and go to the Function tab. It shows the list of functions available.



On choosing **String** from the functions category, we can see only few functions, which take a string as an argument.



In the next chapters, we will see the use of many important functions.

## 18. QlikView – IntervalMatch

QlikView IntervalMatch is a powerful function used to match distinct numeric values to numeric intervals. It is useful in analyzing how the events actually happened versus the planned events. The example of a scenario where it is used is in the assembly lines of the production houses where the belts are planned to run at certain times and for certain duration. However, the actual run can happen at different points in time because of breakdown etc.

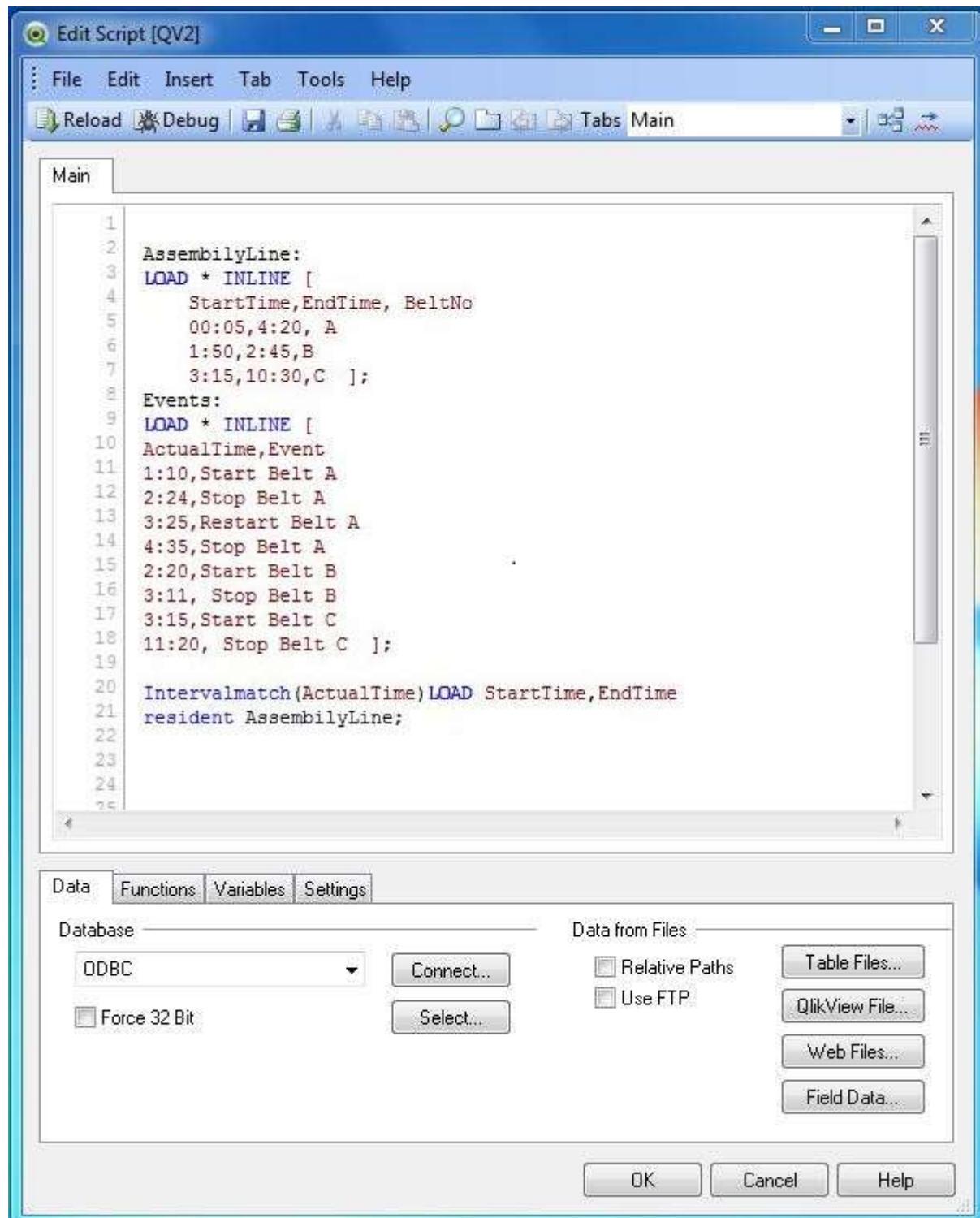
### Example

Consider an assembly line where there are three belts named A, B and C. They are planned to start & stop at specific times of a day. In a given day, we study the actual start and end time and analyze what all happened in that day. For this, we consider two sets of observations as shown below.

```
# Data Set for AssemblyLine.  
StartTime,EndTime, BeltNo  
  
00:05,4:20, A  
1:50,2:45,B  
3:15,10:30,C  
  
# Data set for the events happened.  
ActualTime,Product  
  
1:10,Start Belt A  
2:24,Stop Belt A  
3:25,Restart Belt A  
4:35,Stop Belt A  
2:20,Start Belt B  
3:11, Stop Belt B  
3:15,Start Belt C  
11:20, Stop Belt C
```

## Creating the Script

We open the script editor in a new QlikView document using **Control+E**. The following code creates the required tables as inline data. After creating this script, press **control+R** to reload the data into the QlikView document.



```

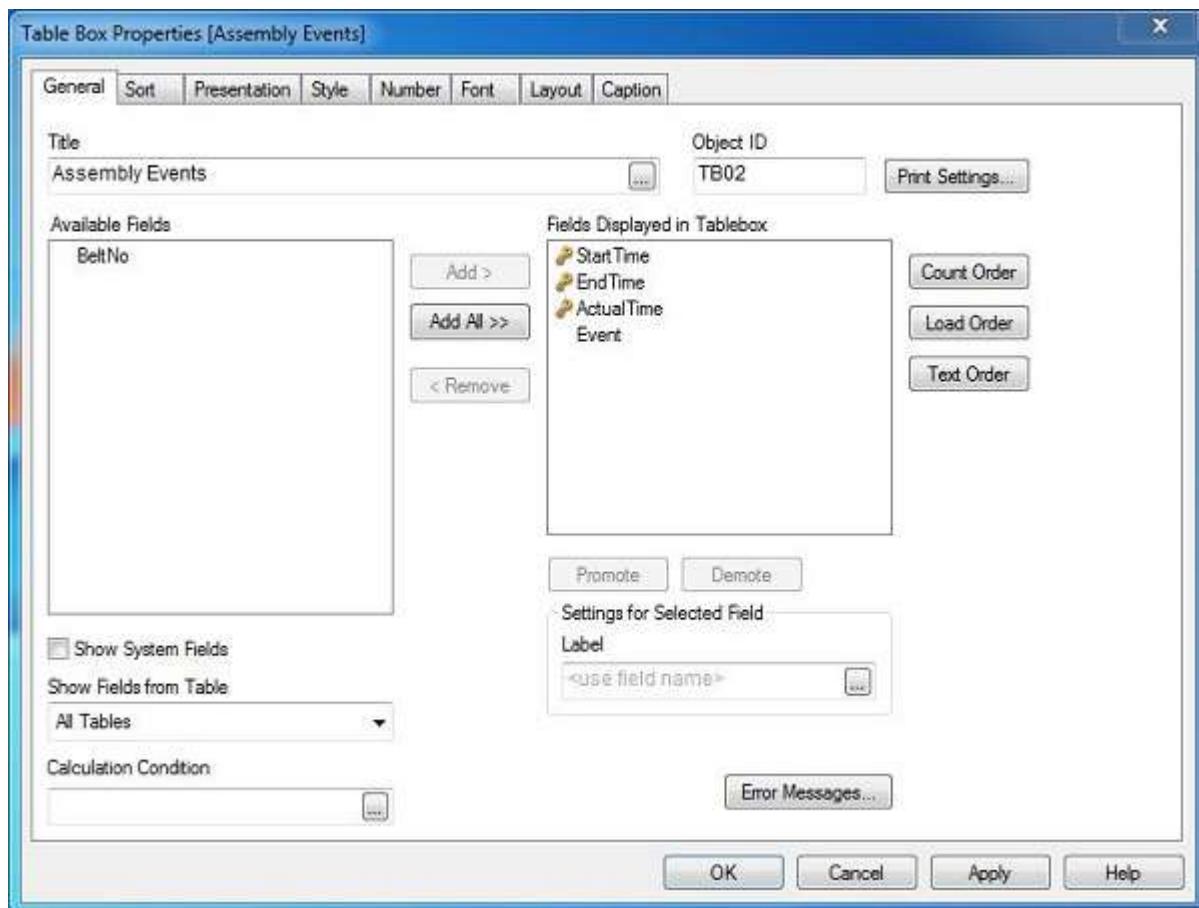
1
2 AssemblyLine:
3 LOAD * INLINE [
4     StartTime,EndTime, BeltNo
5     00:05,4:20, A
6     1:50,2:45, B
7     3:15,10:30, C ];
8
9 Events:
10 LOAD * INLINE [
11     ActualTime,Event
12     1:10,Start Belt A
13     2:24,Stop Belt A
14     3:25,Restart Belt A
15     4:35,Stop Belt A
16     2:20,Start Belt B
17     3:11, Stop Belt B
18     3:15,Start Belt C
19     11:20, Stop Belt C ];
20
21 Intervalmatch(ActualTime) LOAD StartTime,EndTime
22 resident AssemblyLine;
23
24
25

```

The screenshot shows the QlikView Script Editor window titled "Edit Script [QV2]". The main pane displays the QlikView script code. Below the main pane, there is a toolbar with icons for Reload, Debug, and various file operations. The tabs bar shows "Main" is selected. At the bottom of the window, there are tabs for Data, Functions, Variables, and Settings, with "Data" currently selected. The "Database" section contains an ODBC dropdown menu, a "Connect..." button, and a "Force 32 Bit" checkbox. The "Data from Files" section includes checkboxes for "Relative Paths" and "Use FTP", and buttons for "Table Files...", "QlikView File...", "Web Files...", and "Field Data...". At the bottom right are buttons for OK, Cancel, and Help.

## Creating Sheet Object

Let us create a **Table Box** sheet object to show the data generated by the IntervalMatch function. Go to the menu item **Layout -> New Sheet Object -> Table Box**. The following window appears in which we mention the Title of the table and select the required fields to be displayed.



## Showing the Table Box

92

On clicking OK in the above window, a table appears showing the field ActualTime matched to the intervals StartTime and EndTime.

The screenshot shows the QlikView interface with a title bar "QlikView x64 Personal Edition - [QV1]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. Below the menu is a toolbar with various icons. The main area is titled "Main" and contains a table titled "Assembly Events". The table has four columns: StartTime, EndTime, ActualTime, and Event. The data in the table is as follows:

StartTime	EndTime	ActualTime	Event
00:05	4:20	1:10	Start Belt A
00:05	4:20	2:20	Start Belt B
00:05	4:20	2:24	Stop Belt A
00:05	4:20	3:11	Stop Belt B
00:05	4:20	3:15	Start Belt C
00:05	4:20	3:25	Restart Belt A
1:50	2:45	2:20	Start Belt B
1:50	2:45	2:24	Stop Belt A
3:15	10:30	3:15	Start Belt C
3:15	10:30	3:25	Restart Belt A
3:15	10:30	4:35	Stop Belt A
		11:20	Stop Belt C

At the bottom left, it says "For Help, press F1".

# 19. QlikView – Aggregate Functions

QlikView Aggregate functions are used to produce aggregate data from the rows of the table. The functions are applied to the columns when creating the load script. Given below is a sample list of Aggregate functions. We also need to apply the **Group by** clause appropriately when applying the aggregate functions.

- **SUM** gives the sum of the numeric values of the column.
- **AVG** gives the average of the numeric values of the column.
- **MAX** gives the maximum of the numeric values of the column.
- **MIN** gives the minimum of the numeric values of the column.

## Example

Consider the following data stored as product\_sales.csv in the local system. It represents the sales figures for different product lines and product category in a store.

```
Product_Line,Product_category,Quantity,Value
Sporting Goods,Outdoor Recreation,12,5642
Food, Beverages & Tobacco,38,2514
Apparel & Accessories,Clothing,54,2365
Apparel & Accessories,Costumes & Accessories,29,4487
Sporting Goods,Athletics,11,812
Health & Beauty,Personal Care,21,6912
Arts & Entertainment,Hobbies & Creative Arts,58,5201
Arts & Entertainment,Paintings,73,8451
Arts & Entertainment,Musical Instruments,41,1245
Hardware,Tool Accessories,2,456
Home & Garden,Bathroom Accessories,36,241
Food,Drinks,54,1247
Home & Garden,Lawn & Garden,29,5462
Office Supplies,Presentation Supplies,22,577
Hardware,Blocks,53,548
Baby & Toddler,Diapering,19,1247
```

## Creating the Load Script

We open the script editor in a new QlikView document using **Control+E**. The following code creates the required tables as inline data. After creating this script press control+R to reload the data into the QlikView document.

The screenshot shows the QlikView script editor window titled "Edit Script [QV2]". The main pane displays the following QlikView script:

```
1
2
3
4 LOAD Product_Line,
5     Product_category,
6     Quantity,
7     Value
8 FROM
9 [E:\Qlikview\data\product_sales.csv]
10 (txt, codepage is 1252, embedded labels, delimiter is ',', msq);
11
12
13
14
15
16
```

Below the script editor, there is a "Data" configuration dialog. It has tabs for Data, Functions, Variables, and Settings. The Data tab is selected. It contains sections for Database and Data from Files. Under Database, there is an ODBC dropdown menu set to "ODBC", a "Connect..." button, and a "Select..." button. There is also a checkbox for "Force 32 Bit". Under Data from Files, there are checkboxes for "Relative Paths" and "Use FTP", and several browse buttons: "Table Files...", "QlikView File...", "Web Files...", and "Field Data...". At the bottom of the dialog are "OK", "Cancel", and "Help" buttons.

## Creating Sheet Object

Let us create a **Table Box** sheet object to show the data generated by the Aggregate function. Go to the menu **Layout -> New Sheet Object -> Table Box**. The following window appears in which we mention the Title of the table and the select the required fields to be displayed. Clicking OK displays the data from the CSV file in the QlikView Table Box as shown below.

The screenshot shows the QlikView x64 Personal Edition interface with the title bar "QlikView x64 Personal Edition - [QV1]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. Below the menu is a toolbar with various icons. The main area is titled "Main" and contains a table box. The table has the following data:

Product_Line	Product_category	Quantity	Value
Apparel & Accessories	Clothing	54	2365
Apparel & Accessories	Costumes & Accessories	29	4487
Arts & Entertainment	Hobbies & Creative Arts	58	5201
Arts & Entertainment	Musical Instruments	41	1245
Arts & Entertainment	Paintings	73	8451
Baby & Toddler	Diapering	19	1247
Food	Beverages & Tobacco	38	2514
Food	Drinks	54	1247
Hardware	Blocks	53	548
Hardware	Tool Accessories	2	456
Health & Beauty	Personal Care	21	6912
Home & Garden	Bathroom Accessories	36	241
Home & Garden	Lawn & Garden	29	5462
Office Supplies	Presentation Supplies	22	577
Sporting Goods	Athletics	11	812
Sporting Goods	Outdoor Recreation	12	5642

At the bottom left of the main area, it says "For Help, press F1".

## Applying SUM() function

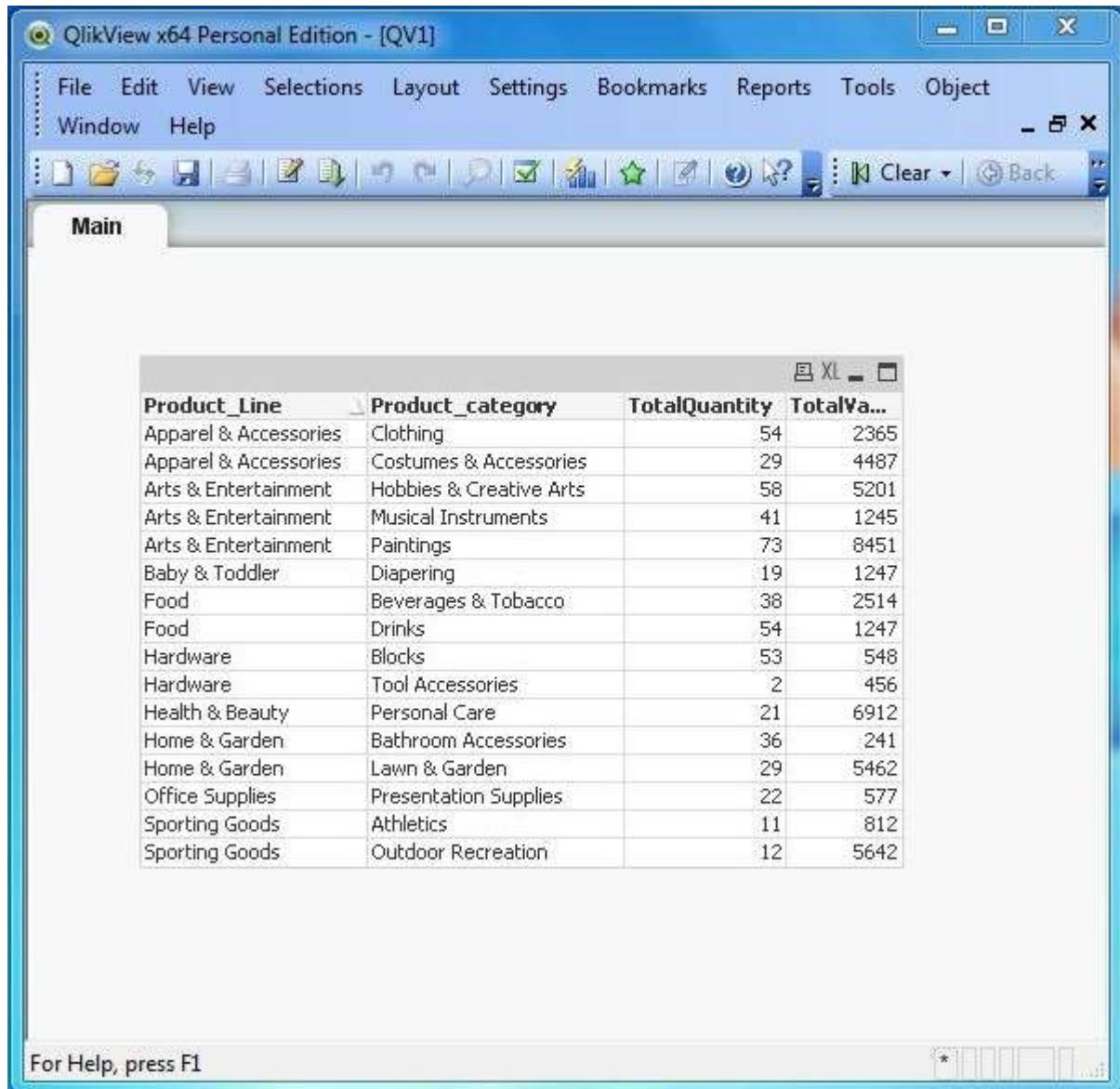
Given below is the load script to find the sum of the sales quantity and sales value across the Product Lines and product categories.

The screenshot shows the 'Edit Script [QV2]' dialog in QlikView. The main area displays the following QlikScript:

```
1
2
3
4 LOAD Product_Line,
5     Product_category,
6     sum(Quantity) as TotalQuantity,
7     Sum(Value) as TotalValue
8 FROM
9     [E:\Qlikview\data\product_sales.csv]
10    (txt, codepage is 1252, embedded labels, delimiter is ',', msq)
11    Group by Product_Line,Product_category;
12
13
14
15
16
```

Below the script, there are tabs for Data, Functions, Variables, and Settings. Under the Data tab, the Database section is set to ODBC, with options for Connect..., Select..., Force 32 Bit, and Relative Paths. The Data from Files section includes buttons for Table Files..., QlikView File..., Web Files..., and Field Data... . At the bottom are OK, Cancel, and Help buttons.

Click OK and press **Control+R** to reload the data into QlikView document. Now follow the same steps as given above in - **Creating Sheet Objects** to create a QlikView Table Box for displaying the result of the script as shown below.



The screenshot shows the QlikView interface with a table box titled 'Main' containing the following data:

Product_Line	Product_category	TotalQuantity	TotalValue
Apparel & Accessories	Clothing	54	2365
Apparel & Accessories	Costumes & Accessories	29	4487
Arts & Entertainment	Hobbies & Creative Arts	58	5201
Arts & Entertainment	Musical Instruments	41	1245
Arts & Entertainment	Paintings	73	8451
Baby & Toddler	Diapering	19	1247
Food	Beverages & Tobacco	38	2514
Food	Drinks	54	1247
Hardware	Blocks	53	548
Hardware	Tool Accessories	2	456
Health & Beauty	Personal Care	21	6912
Home & Garden	Bathroom Accessories	36	241
Home & Garden	Lawn & Garden	29	5462
Office Supplies	Presentation Supplies	22	577
Sporting Goods	Athletics	11	812
Sporting Goods	Outdoor Recreation	12	5642

## Applying AVG() function

Given below is the load script to create the average of the sales quantity and sales value across each Product Line.

```
# Average sales of Quantity and value in each Product Line.

LOAD Product_Line,
    avg(Quantity),
    avg(Value)
FROM
[E:\Qlikview\data\product_sales.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq)
```

**Group by Product\_Line;**

Click OK and press **Control+R** to reload the data into QlikView document. Now follow the same steps as given above in - **Creating Sheet Objects** to create a QlikView Table Box for displaying the result of the script as shown below.

The screenshot shows the QlikView interface with a window titled "QlikView x64 Personal Edition - [QV1]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. Below the menu is a toolbar with various icons. The main area is labeled "Main" and contains a table titled "Average Sales". The table has three columns: "Product\_Line", "AverageQuantity", and "AverageValue". The data is as follows:

Product_Line	AverageQuantity	AverageValue
Sporting Goods	11.5	3,227
Baby & Toddler	19	1,247
Health & Beauty	21	6,912
Office Supplies	22	577
Hardware	27.5	502
Home & Garden	32.5	2,852
Apparel & Accessories	41.5	3,426
Food	46	1,881
Arts & Entertainment	57.3	4,966

At the bottom of the screen, there is a message "For Help, press F1".

## Applying MAX() & MIN() function

Given below is the load script to create the maximum and minimum of the sales quantity across each Product Line.

```
# Maximum and Minimum sales in each product Line.

LOAD Product_Line,
    max(Quantity) as MaxQuantity,
    min(Quantity) as MinQuantity
FROM
[E:\Qlikview\data\product_sales.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq)
Group by Product_Line;
```

Click OK and **Control+R** to reload the data into QlikView document. Now follow the same steps as above in - **Creating Sheet Objects** to create a QlikView Table Box for displaying the result of the script as shown below.

QlikView x64 Personal Edition - [QV1]

Main

Product_Line	MaxQuantity	MinQuantity
Apparel & Accessories	54	29
Arts & Entertainment	73	41
Baby & Toddler	19	19
Food	54	38
Hardware	53	2
Health & Beauty	21	21
Home & Garden	36	29
Office Supplies	22	22
Sporting Goods	12	11

For Help, press F1

## 20. QlikView – Match Function

The **Match()** function in QlikView is used to match the value of a string or expression with the data value present in a column. It is similar to the **in** function that we see in SQL language. It is useful to fetch rows containing specific strings and it also has an extension in form of **wildmatch()** function.

Let us consider the following data as input file for the examples illustrated below.

```
Product_Id,Product_Line,Product_category,Product_Subcategory
1,Sporting Goods,Outdoor Recreation,Winter Sports & Activities
2,Food, Beverages & Tobacco,Food Items,Fruits & Vegetables
3,Apparel & Accessories,Clothing,Uniforms
4,Sporting Goods,Athletics,Rugby
5,Health & Beauty,Personal Care
6,Arts & Entertainment,Hobbies & Creative Arts,Musical Instruments
7,Arts & Entertainment,Hobbies & Creative Arts,Orchestra Accessories
8,Arts & Entertainment,Hobbies & Creative Arts,Crafting Materials
9,Hardware,Tool Accessories,Power Tool Batteries
10,Home & Garden,Bathroom Accessories,Bath Caddies
11,Food, Beverages & Tobacco,Food Items,Frozen Vegetables
12,Home & Garden,Lawn & Garden,Power Equipment
13,Office Supplies,Presentation Supplies,Display
14,Hardware,Tool Accessories,Jigs
15,Baby & Toddler,Diapering,Baby Wipes
```

## Load Script with Match() Function

The following script shows the Load script, which reads the file named product\_categories.csv. We search the field **Product\_Line** for values matching with strings 'Food' and 'Sporting Goods'.

The screenshot shows the 'Edit Script [QV2]' dialog in QlikView. The main area contains the following QlikView script:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
LOAD Product_Id,
      Product_Line,
      Product_category,
      Product_Subcategory
FROM
[E:\Qlikview\data\product_categories.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq)
where match(Product_Line,'Food','Sporting Goods')
```

Below the script, there are tabs for Data, Functions, Variables, and Settings. Under the Data tab, there are sections for Database and Data from Files. The Database section includes dropdowns for ODBC and Force 32 Bit, and buttons for Connect... and Select... . The Data from Files section includes checkboxes for Relative Paths and Use FTP, and buttons for Table Files..., QlikView File..., Web Files..., and Field Data... . At the bottom are OK, Cancel, and Help buttons.

## Creating Sheet Object

Let us create a **Table Box** sheet object to show the data generated by the match function. Go to the menu **Layout -> New Sheet Object -> Table Box**. The following window appears in which we mention the Title of the table and then select the required fields to be displayed. Clicking OK displays the data from the CSV file in the QlikView Table Box as shown below.

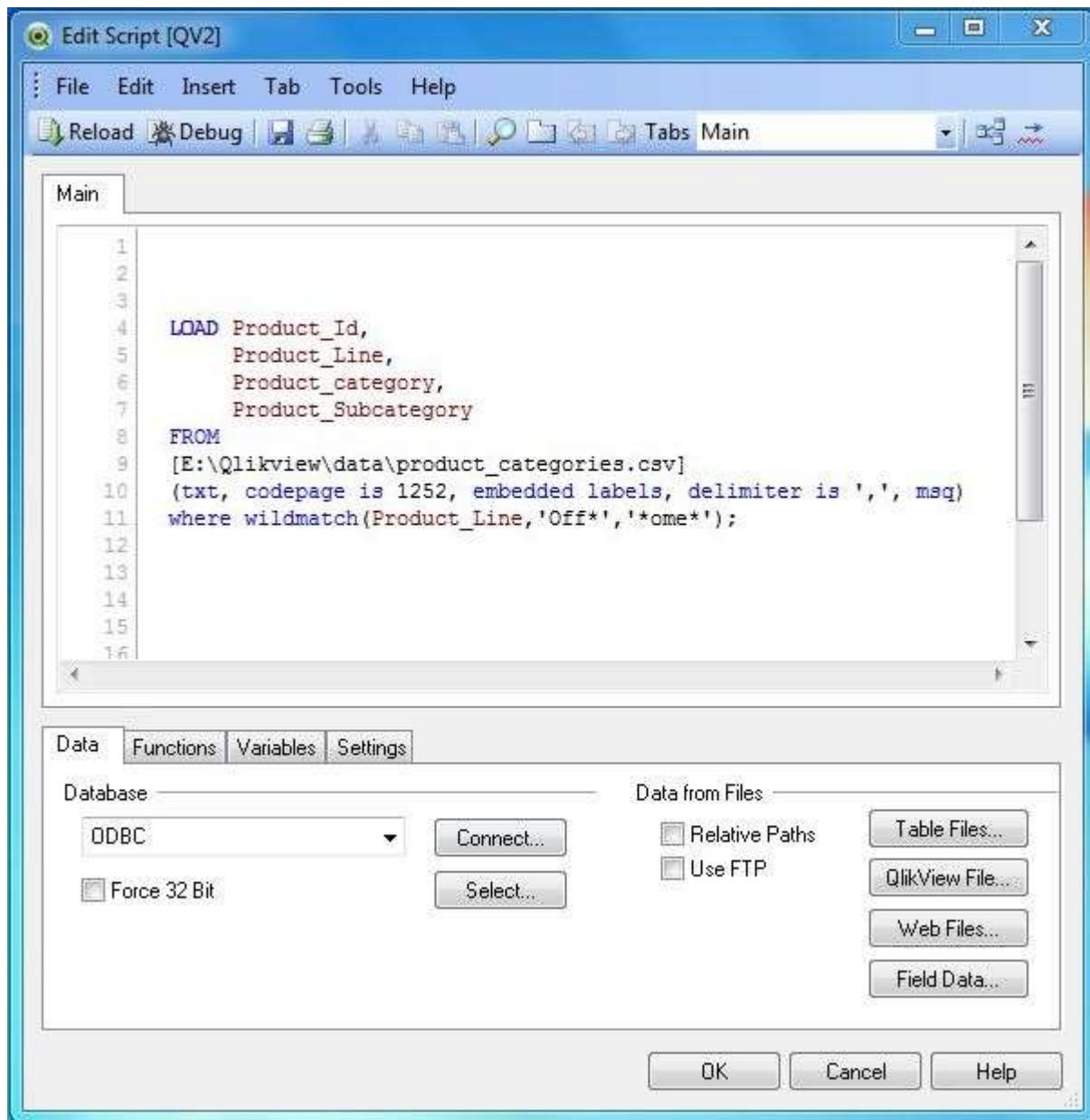
The screenshot shows the QlikView x64 Personal Edition interface. The title bar reads "QlikView x64 Personal Edition - [QV1]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. Below the menu is a toolbar with various icons. The main workspace is titled "Main". In the center, there is a "Table Box" visualization titled "Matched Products". The table has four columns: Product\_Id, Product\_Line, Product\_category, and Product\_Subcategory. The data is as follows:

Product_Id	Product_Line	Product_category	Product_Subcategory
1	Sporting Goods	Outdoor Recreation	Winter Sports & Activities
2	Food	Beverages & Tobacco	Food Items
4	Sporting Goods	Athletics	Rugby
11	Food	Beverages & Tobacco	Food Items

At the bottom left of the workspace, it says "For Help, press F1".

## Load Script with Wildmatch() Function

The **wildmatch()** function is an extension of match() function in which we can use wildcards as part of the strings used to match the values with values in the fields being searched for. We search for the strings 'Off\*', '\*ome\*'.



## Creating Sheet Object

Let us create a **Table Box** sheet object to show the data generated by the wildmatch function. Go to the menu item **Layout -> New Sheet Object -> Table Box**. The following window appears in which we mention the Title of the table and then select the required fields to be displayed. Clicking OK displays the data from the CSV file in the QlikView Table Box as shown below.

The screenshot shows the QlikView x64 Personal Edition interface. The title bar reads "QlikView x64 Personal Edition - [QV1]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. Below the menu is a toolbar with various icons. The main workspace is titled "Main". A table box is displayed with the title "Matched Products". The table has three columns: "Product\_Id", "Product\_Line", "Product\_category", and "Product\_Subcategory". The data is as follows:

Product_Id	Product_Line	Product_category	Product_Subcategory
10	Home & Garden	Bathroom Accessories	Bath Caddies
12	Home & Garden	Lawn & Garden	Power Equipment
13	Office Supplies	Presentation Supplies	Display

At the bottom left of the workspace, it says "For Help, press F1".

## 21. QlikView – Rank Function

The **Rank()** function in QlikView is used to display the rank of the values in a field as well as return rows with specific rank value. So it is used in two scenarios. First scenario is in QlikView charts to display the ranks of the values in the field and second is in Aggregate function to display only the rows, which have a specific rank value.

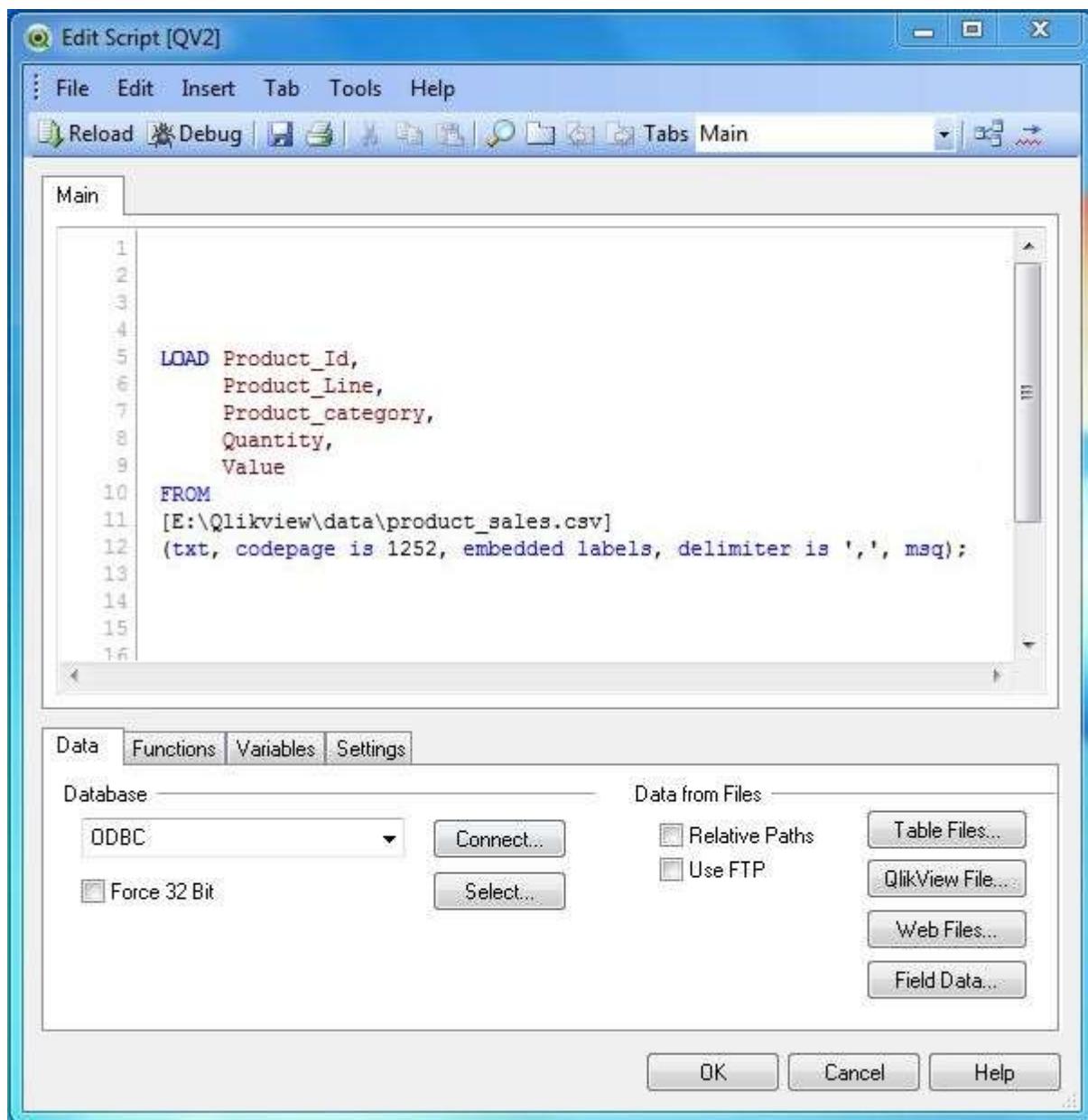
### Input Data

The data used in the examples describing Rank function is given below. You can save this as a .csv file in a path in your system where it is accessible by QlikView.

```
Product_Id,Product_Line,Product_category,Quantity,Value
1,Sporting Goods,Outdoor Recreation,12,5642
2,Food, Beverages & Tobacco,38,2514
3,Apparel & Accessories,Clothing,54,2365
4,Apparel & Accessories,Costumes & Accessories,29,4487
5,Sporting Goods,Athletics,11,812
6,Health & Beauty,Personal Care,21,6912
7,Arts & Entertainment,Hobbies & Creative Arts,58,5201
8,Arts & Entertainment,Paintings,73,8451
9,Arts & Entertainment,Musical Instruments,41,1245
10,Hardware,Tool Accessories,2,456
11,Home & Garden,Bathroom Accessories,36,241
12,Food,Drinks,54,1247
13,Home & Garden,Lawn & Garden,29,5462
14,Office Supplies,Presentation Supplies,22,577
15,Hardware,Blocks,53,548
16,Baby & Toddler,Diapering,19,1247
17,Baby & Toddler,Toys,9,257
18,Home & Garden,Pipes,81,1241
19,Office Supplies,Dispaly Borad,29,2177
```

## Load Script

The above data is loaded to the QlikView memory by using the script editor. Open the Script editor from the File menu or press **Control+E**. Choose the **Table Files** option from the **Data from Files** tab and browse for the file containing the above data. Click **OK** and press **Control+R** to load the data into QlikView's memory.



## Creating Chart with Rank() Function

Next, we follow the steps given below to create a chart, which shows the rank of the filed **Value** described with respect to the dimension Product\_Line.

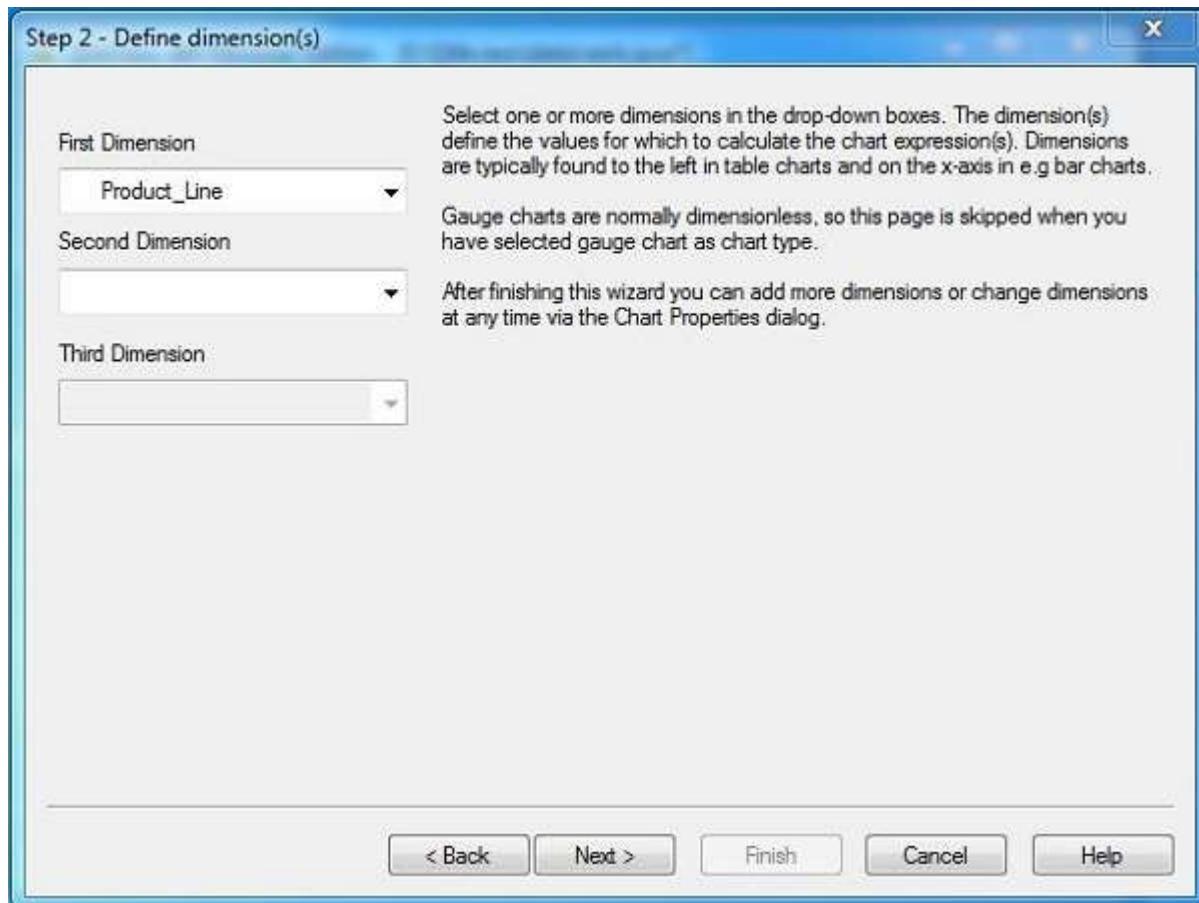
### Select the Chart Type

Click on the Chart wizard and choose the option **straight table** as the chart type. Click **Next**.



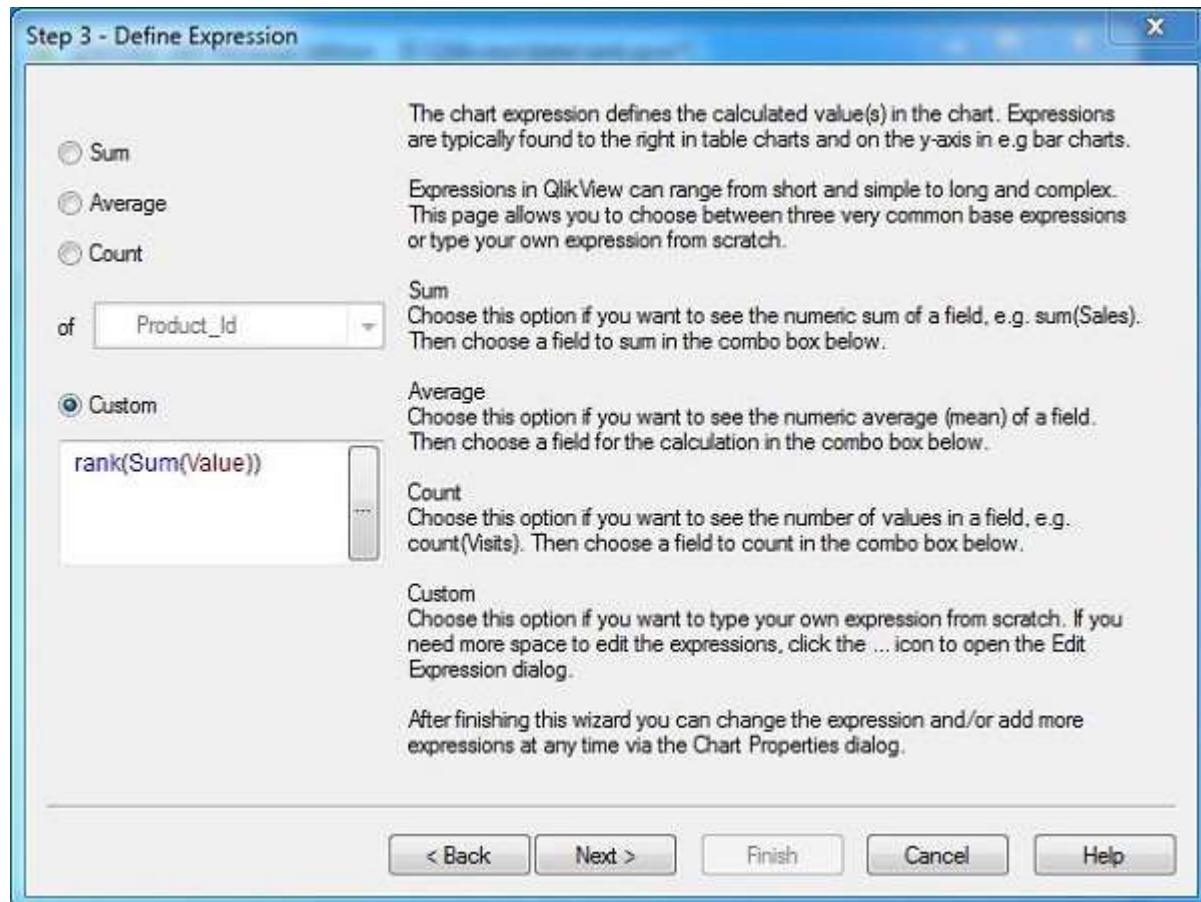
## Select the Chart Dimension

From the **First Dimension** drop down list, choose Product\_Line as dimension. Click **Next**.



## Define the Chart Expression

In the custom expression field, mention the rank expression as shown below. Here we are considering the numeric field named **Value**, which represents the Sales value for each category under each Product Line. Click **Next**.



## Displaying the Chart

On clicking Finish in the above step, the following chart appears which shows the rank of the sales value of each Product Line.

rank(Sum(Value))	
Product_Line	rank(Sum(Value))
Sporting Goods	5
Food	6
Apparel & Accessories	4
Health & Beauty	3
Arts & Entertainment	1
Hardware	9
Home & Garden	2
Office Supplies	7
Baby & Toddler	8

## Using Aggregate Function with Rank

The aggregate functions like - max, min etc. can take rank as an argument to return rows satisfying certain rank values. We consider the following expression to be out in the script editor, which will give the rows containing highest sales under each Product line.

```
# Load the records with highest sales value for each product line.
LOAD Product_Line,
    max(Value,1)
FROM
[E:\Qlikview\data\product_sales.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq)
group by Product_Line;
```

## Creating Sheet Object

Let us create a **Table Box** sheet object to show the data generated by the above given script. Go to the menu **Layout -> New Sheet Object -> Table Box**. The following window appears in which we mention the Title of the table and select the required fields to be displayed. Clicking OK displays the data from the CSV file in the QlikView Table Box as shown below.

The screenshot shows the QlikView x64 Personal Edition interface. The title bar reads "QlikView x64 Personal Edition - [E:\Qlikview\data\rank.qvw\*]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. The toolbar below the menu bar contains various icons for file operations like Open, Save, Print, and zoom. The main workspace is titled "Main" and contains a "Table Box" object. The table has two columns: "Product\_Line" and "max(Value,1)". The data is as follows:

Product_Line	max(Value,1)
Hardware	548
Baby & Toddler	1247
Office Supplies	2177
Food	2514
Apparel & Accessories	4487
Home & Garden	5462
Sporting Goods	5642
Health & Beauty	6912
Arts & Entertainment	8451

At the bottom of the workspace, there is a status bar with the text "For Help, press F1" and the date and time "10/31/2015 7:49:56 AM\*".

## 22. QlikView – Peek Function

The **peek()** function in QlikView is used to fetch the value of a field from a previous record and use it in calculations.

### Input Data

Let us consider the monthly sales figure as shown below. Save the data with file name `monthly_sales.csv`.

```
Month,Sales Volume
March,2145
April,2458
May,1245
June,5124
July,7421
August,2584
September,5314
October,7846
November,6532
December,4625
January,8547
February,3265
```

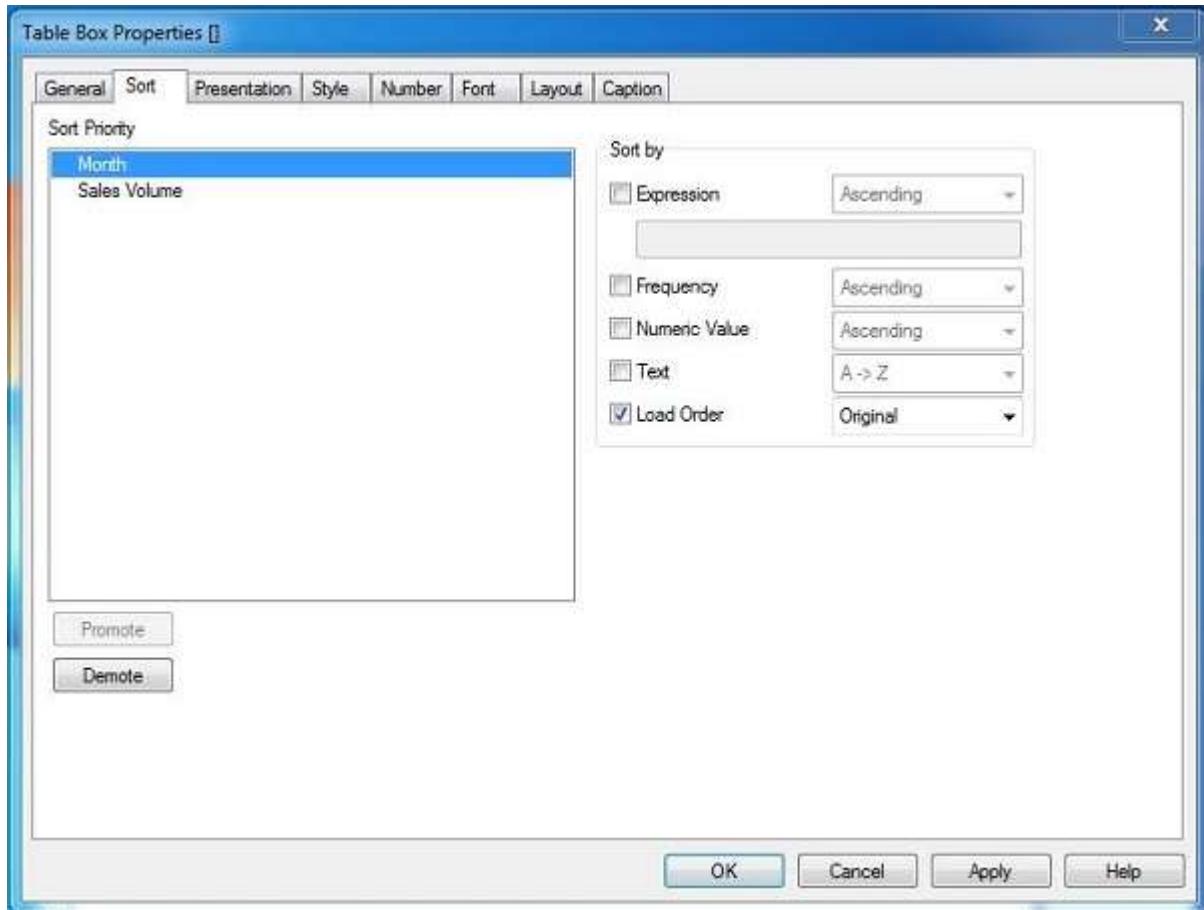
### Load Script

The above data is loaded to QlikView memory by using the script editor. Open the Script editor from the File menu or press **Control+E**. Choose the **Table Files** option from the **Data from Files** tab and browse for the file containing the above data. Edit the load script to add the following code. Click **OK** and click **Control+R** to load the data into QlikView's memory.

```
LOAD Month,
      [Sales Volume],
      peek('Sales Volume') as Prevmonth
FROM
[C:\Qlikview\data\monthly_sales.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);
```

## Creating Sheet Object

Let us create a **Table Box** sheet object to show the data generated by the above script. Go to the menu item **Layout -> New Sheet Object -> Table Box**. The following window appears in which we mention the Title of the table and select the required fields to be displayed. Clicking OK displays the data from the csv file in the QlikView Table Box as shown below. Also set the sort order as shown below to get the result in the same order of the field **Month** as it is in the source.



On completing the above steps and clicking **Finish**, we get the Table box showing the data as given below.

Month	Sales Volume	Prevmonth
March	2145	-
April	2458	2145
May	1245	2458
June	5124	1245
July	7421	5124
August	2584	7421
September	5314	2584
October	7846	5314
November	6532	7846
December	4625	6532
January	8547	4625
February	3265	8547

## Using peek() Function in Calculations

The **peek()** can be used in calculations involving other columns. Let us display the percentage change for sales volume for each month. The following script achieves this result.

```

LOAD
Month, [Sales Volume],
peek('Sales Volume') as Prevvolume,
(([Sales Volume]-peek('Sales Volume')))/peek('Sales Volume')*100 as Difference
FROM
[C:\Qlikview\data\monthly_sales.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);

```

## Creating Sheet Object

Let us create a **Table Box** sheet object to show the data generated by the above script. Go to the menu item **Layout -> New Sheet Object -> Table Box**. The following window appears in which we mention the Title of the table and select the required fields to be

displayed. Clicking OK displays the data from the CSV file in the QlikView Table Box as shown below.

The screenshot shows the QlikView x64 Personal Edition interface. The title bar reads "QlikView x64 Personal Edition - [C:\Qlikview\files\peek\_function.qvw]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. Below the menu is a toolbar with various icons. The main area is titled "Main" and contains a table box. The table has a header row with columns: Month, Sales Volume, Prevvolume, and Difference. The data rows show monthly sales volumes and their differences from the previous month. The table is currently sorted by Sales Volume in descending order. At the bottom left of the table box is the text "For Help, press F1". At the bottom right is the date and time "11/1/2015 12:34:02 PM".

Month	Sales Volume	Prevvolume	Difference
March	2145	-	-
April	2458	2145	14.59
May	1245	2458	-49.35
June	5124	1245	311.6
July	7421	5124	44.83
August	2584	7421	-65.18
September	5314	2584	105.7
October	7846	5314	47.65
November	6532	7846	-16.75
December	4625	6532	-29.19
January	8547	4625	84.8
February	3265	8547	-61.8

## 23. QlikView – RangeSum Function

The **RangeSum()** function in QlikView is used to do a selective sum on chosen fields which is not easily achieved by the sum function. It can take expressions containing other functions as its arguments and return the sum of those expressions.

### Input Data

Let us consider the monthly sales figure as shown below. Save the data with file name monthly\_sales.csv.

```
Month,Sales Volume
March,2145
April,2458
May,1245
June,5124
July,7421
August,2584
September,5314
October,7846
November,6532
December,4625
January,8547
February,3265
```

### Load Script

The above data is loaded to QlikView memory by using the script editor. Open the Script editor from the File menu or press **Control+E**. Choose the **Table Files** option from the **Data from Files** tab and browse for the file containing the above data. Edit the load script to add the following code. Click **OK** and click **Control+R** to load the data into QlikView's memory.

```
LOAD
Month, [Sales Volume]
FROM
[C:\Qlikview\data\monthly_sales.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);
```

## Applying RangeSum() Function

With the above data loaded into QlikView's memory, we edit the script to add a new column, which will give a rolling sum of the month wise sales volume. For this, we also take the help of the peek function discussed in the earlier chapter to hold the value of the previous record and add it to the sales volume of the current record. The following script achieves the result.

```
LOAD
Month, [Sales Volume],
rangeSum([Sales Volume],peek('Rolling')) as Rolling
FROM
[C:\Qlikview\data\monthly_sales.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);
```

## Creating Sheet Object

Let us create a **Table Box** sheet object to show the data generated by the above given script. Go to the menu **Layout -> New Sheet Object -> Table Box**.

The following window appears in which we mention the Title of the table and select the required fields to be displayed. Clicking OK displays the data from the CSV file in the QlikView Table Box as shown below.

The screenshot shows the QlikView x64 Personal Edition interface with the title bar "QlikView x64 Personal Edition - [C:\Qlikview\files\peek\_function.qvw]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and zoom. Below the toolbar is a ribbon bar with tabs like Main, Data, and View. A status bar at the bottom left says "For Help, press F1" and the date and time "11/1/2015 2:51:08 PM". The main area displays a table titled "Main" with three columns: Month, Sales Volume, and Rolling. The data is as follows:

Month	Sales Volume	Rolling
March	2145	2145
April	2458	4603
May	1245	5848
June	5124	10972
July	7421	18393
August	2584	20977
September	5314	26291
October	7846	34137
November	6532	40669
December	4625	45294
January	8547	53841
February	3265	57106

## 24. QlikView – Documents

QlikView documents are the files that contain all the objects used for the data presentation and analysis. It contains the sheets, variables, data model, source-data connection details, and even the data that is loaded after pulling it from the source.

### Document Properties

We can quickly find out the basic information of a QlikView document. Click on **Help -> document Support Info**. Given below is a sample output.

The screenshot shows the 'Document Support Information' dialog box. It displays various properties of a QlikView document. The properties are categorized into sections: Document Info, Object Cache Info, QlikView Info, and INI Values. The 'Document Info' section shows the document path as C:\Qlikview\files\document\_properties.qvw and its local connection. Other sections show settings like object cache limit (806.73 MB), QlikView edition (64-bit Edition), andINI values for QlikView components.

Document Info	
Document	C:\Qlikview\files\document_properties.qvw
Connection	Local
DecimalSep	Doc="."(2E) Sys="."(2E)
ThousandSep	Doc=","(2C) Sys=","(2C)
ListSep	Doc=","(2C) Sys=","(2C)
MoneyDecimalSep	Doc="."(2E) Sys="."(2E)
MoneyThousandSep	Doc=","(2C) Sys=","(2C)
MoneyFmt	Doc="##,##0.00;(##,##0.00)" Sys="##,##0.00;(##,##0.00)"
TimeFmt	Doc="h:mm:ss TT" Sys="h:mm:ss TT"
DateFmt	Doc="M/D/YYYY" Sys="M/D/YYYY"
TimestampFmt	Doc="M/D/YYYY h:mm:ss[.fff] TT" Sys="M/D/YYYY h:mm:ss[.fff] TT"
Document File Format	0 (6000)
Saved in Version	11.20.12904.0409
Client NT Name	
Access	ADMIN
License	-
Document License Status	ALLOWED
Object Cache Info	
Object Cache Limit (MB)	806.73
QlikView Info	
Client Build Number	11.20.12904.0409
Server Build Number	-
QlikTech Product	QlikView 64-bit Edition (x64)
License Key	[Personal Edition]
PE Recoveries Remaining	4
CPU Target	x64
INI Values	
CanDynamicUpdate	1
CurrentQVConnect32Loc	C:\Program Files (x86)\QlikView\QvConnect32.EXE
CurrentQVConnect64Loc	C:\Program Files\QlikView\QvConnect64.EXE

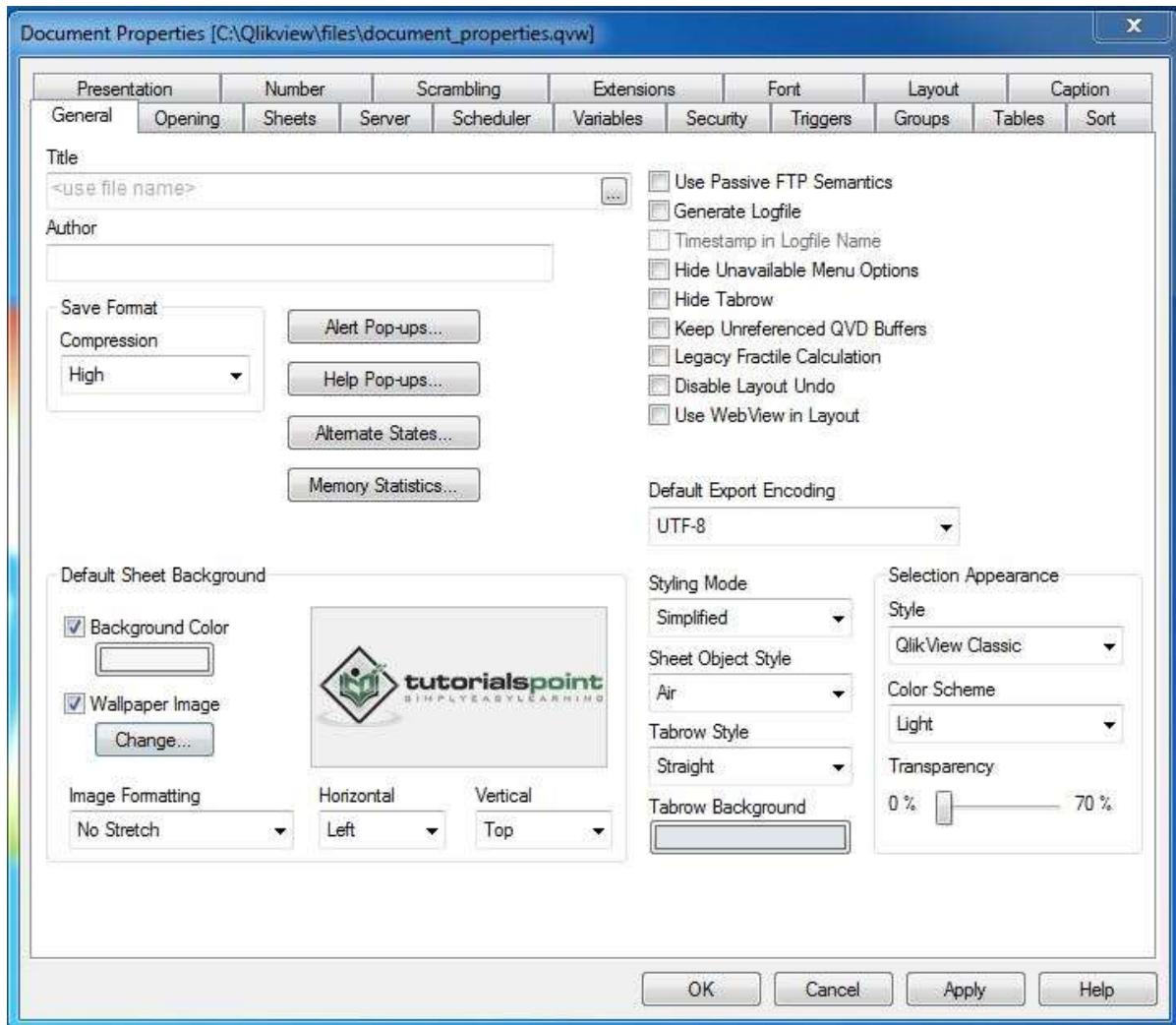
Copy to Clipboard

Close

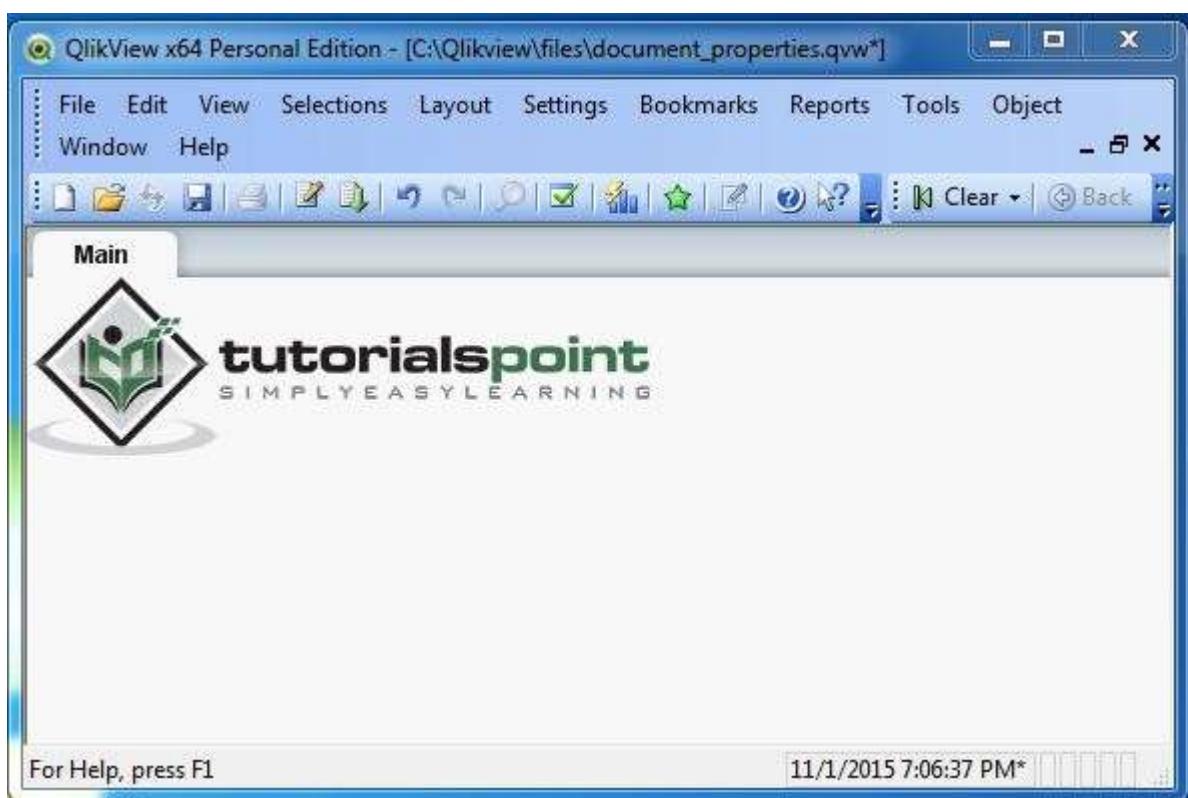
Help

## Setting Document Background Image

We can set an image as the background image for a document using the check box **Wallpaper Image** check box under the **General** tab. We choose an image and align it at the left top position using the dropdown buttons.



The following screen appears on selecting the above options.



## Sheet Objects

The QlikView document contains various Sheet objects, which can be moved around by dragging them and placed anywhere in the document. Let us create two sheet objects, a **Table box** and a **Statistics Box**. You can follow the earlier chapters where we have already learnt to create sheet objects. In addition, we are using the file Product\_sales.csv, which is mentioned here.

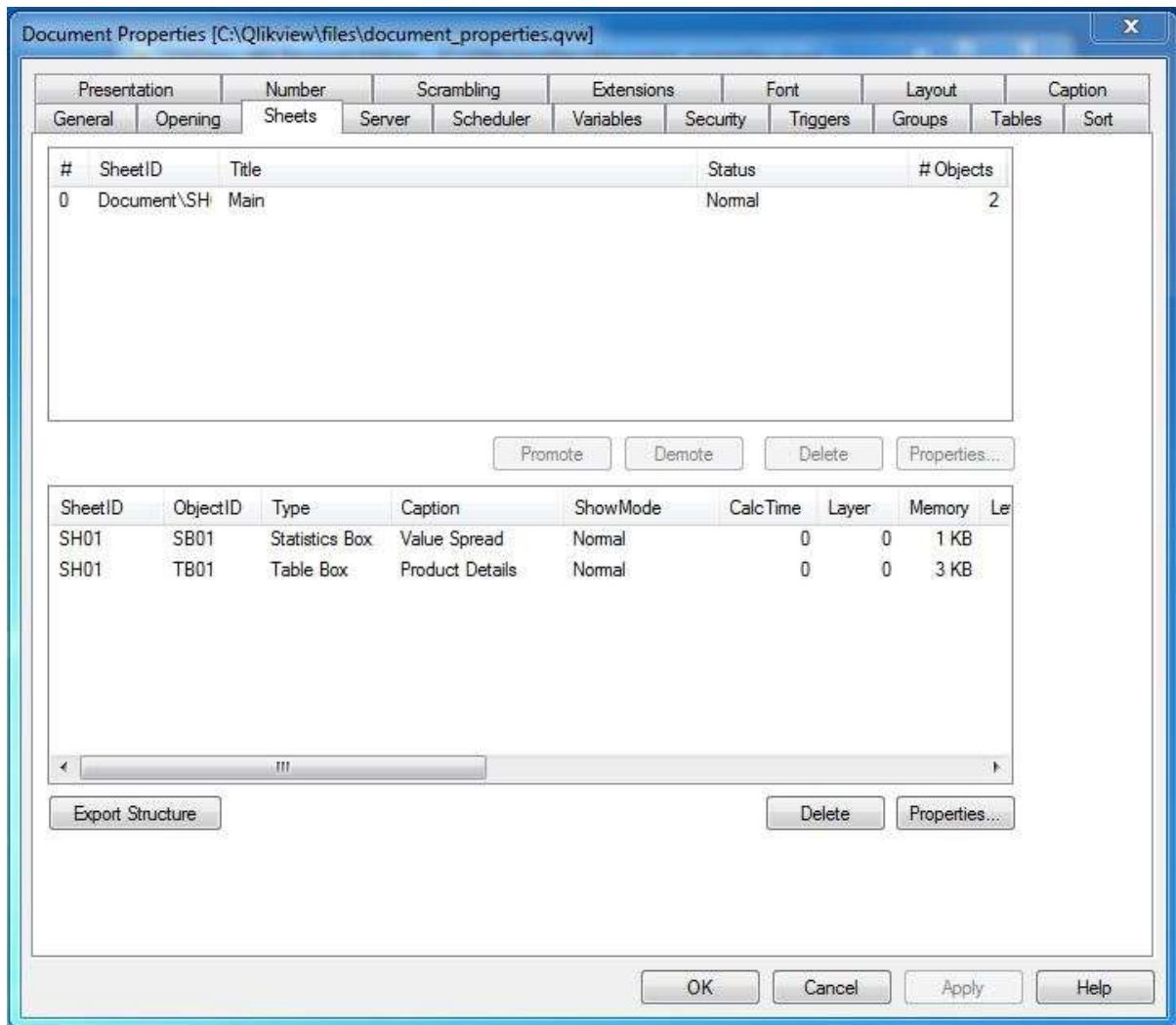
The screenshot shows the QlikView interface with the title bar "QlikView x64 Personal Edition - [C:\Qlikview\files\document\_properties.qvw]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. The toolbar below has various icons for file operations like Open, Save, Print, and search. The main area is titled "Main" and features the "tutorials point SIMPLY EASY LEARNING" logo. Below the logo is a table titled "Product Details" with columns: Product\_category, Product\_Line, and Value. The table lists 19 products from categories like Athletics, Bathrooms, Beverages, etc., with their respective values. To the right of the table is a "Value Spread" box containing statistical summary data:

Total count	19
Sum	51082
Average	2,688.526
Min	241
Max	8451

At the bottom left is a help message "For Help, press F1" and at the bottom right is the date and time "11/1/2015 7:27:15 PM".

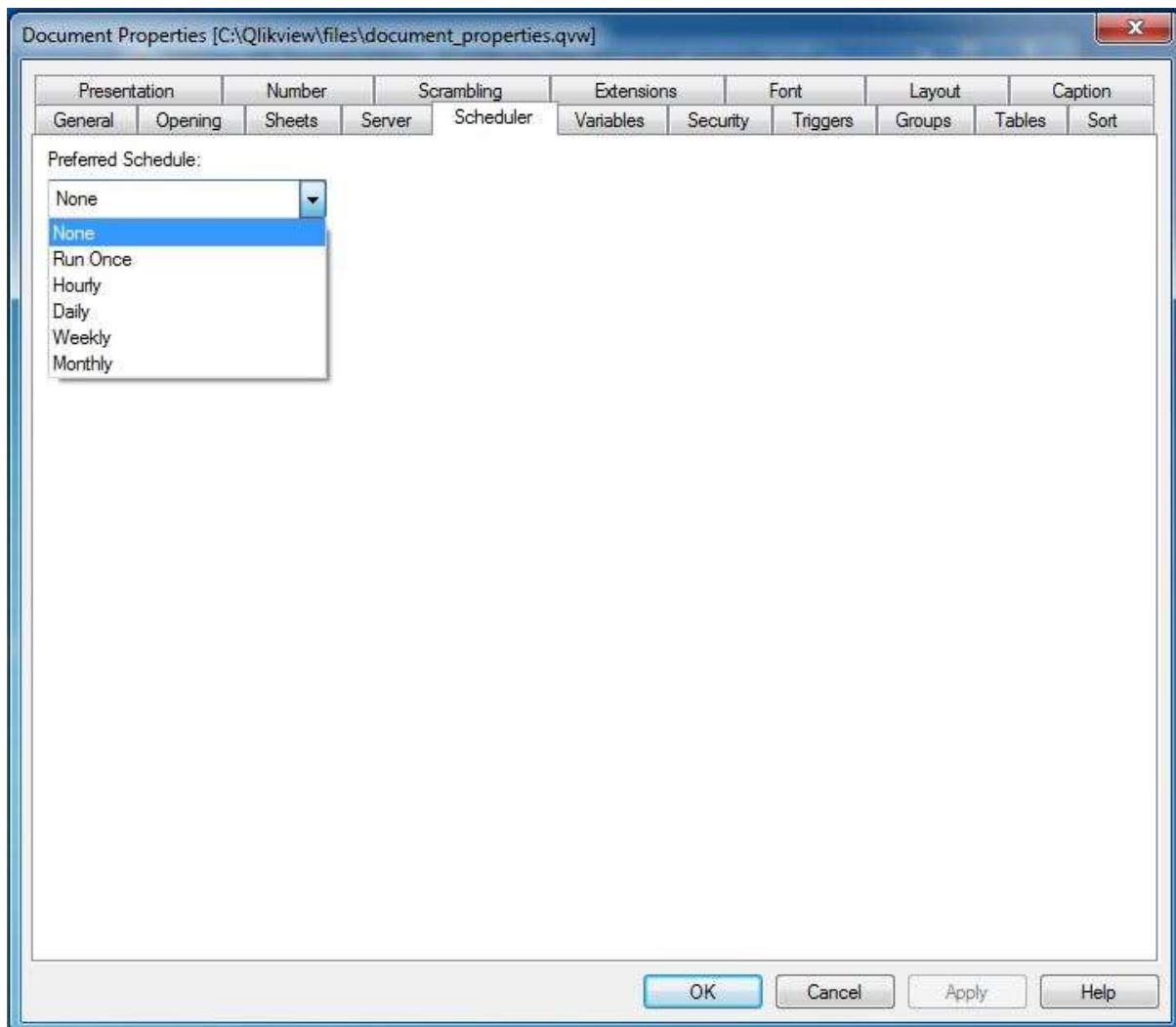
Details of the Sheet objects can be seen using the "Sheets" tab. It shows all the sheets contained in the document and for each sheet, the sheet objects are shown. Both the

sheets and sheet objects have unique IDs. We can also edit various properties of these objects from this tab itself.



## Scheduling a Document

A QlikView document can be scheduled to refresh at some desired intervals. This is done using the **Schedule** tab available under the Document properties window.



## 25. QlikView – List Box

A **list box** represents the list of all the values of a specific field. Selecting a value in list box highlights the related values in other sheet objects. This helps in faster visual analysis. It is also very useful to follow a drill down path among various sheet objects. It also has a search feature, which allows to search for specific values in the list box which is very helpful for a very long list of values.

### Input Data

Let us consider the following input data, which represents the sales figure of different product lines and product categories.

```
Product_Line,Product_category,Value
Sporting Goods,Outdoor Recreation,5642
Food, Beverages & Tobacco,2514
Apparel & Accessories,Clothing,2365
Apparel & Accessories,Costumes & Accessories,4487
Sporting Goods,Athletics,812
Health & Beauty,Personal Care,6912
Arts & Entertainment,Hobbies & Creative Arts,5201
Arts & Entertainment,Paintings,8451
Arts & Entertainment,Musical Instruments,1245
Hardware,Tool Accessories,456
Home & Garden,Bathroom Accessories,241
Food,Drinks,1247
Home & Garden,Lawn & Garden,5462
Office Supplies,Presentation Supplies,577
Hardware,Blocks,548
Baby & Toddler,Diapering,1247
Baby & Toddler,Toys,257
Home & Garden,Pipes,1241
Office Supplies,Dispaly Borad,2177
```

### Load Script

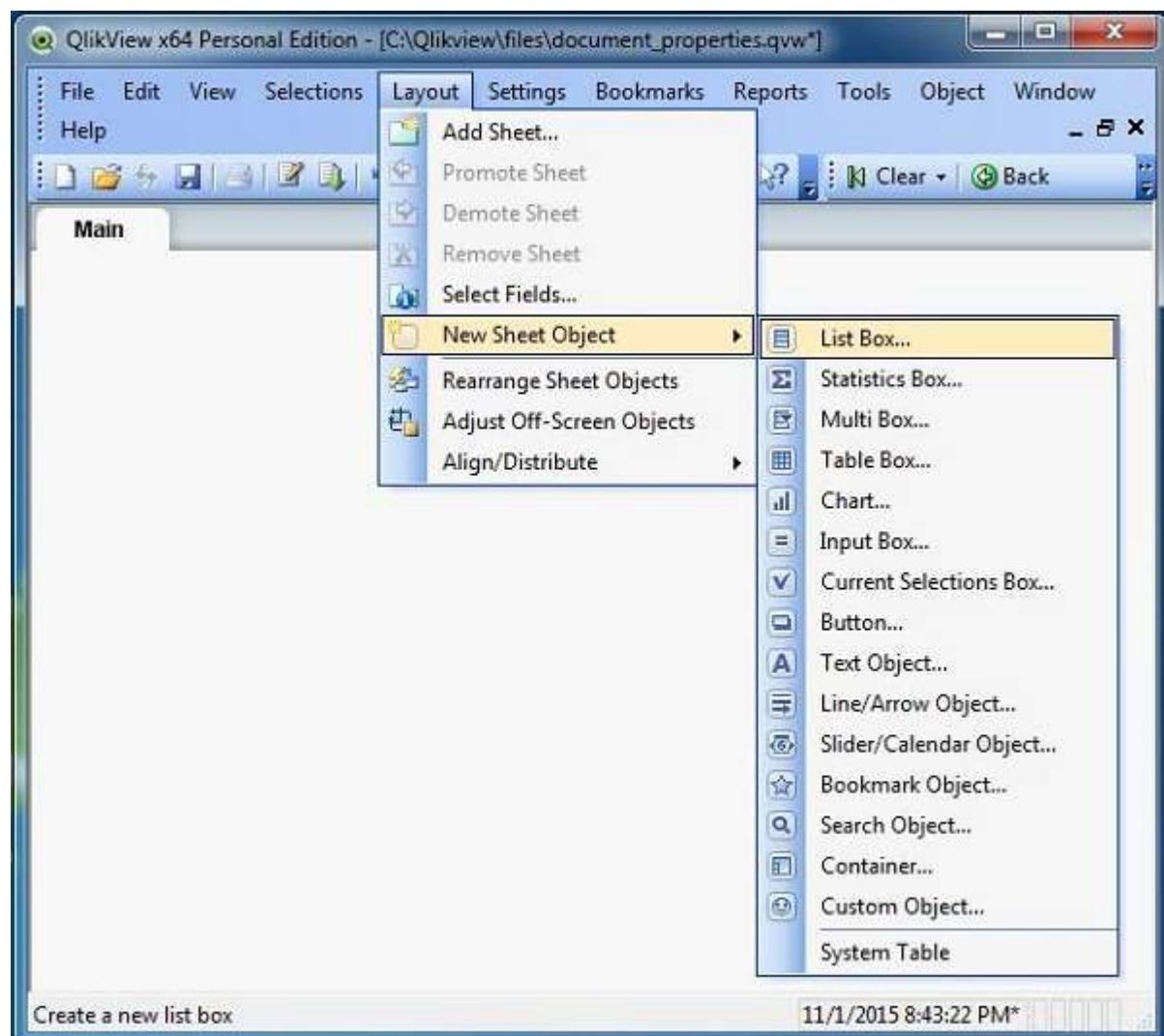
The above data is loaded to QlikView memory by using the script editor. Open the Script editor from the File menu or press **Control+E**. Choose the **Table Files** option from the **Data from Files** tab and browse for the file containing the above data. Edit the load

script to add the following code. Click **OK** and click **Control+R** to load the data into QlikView's memory.

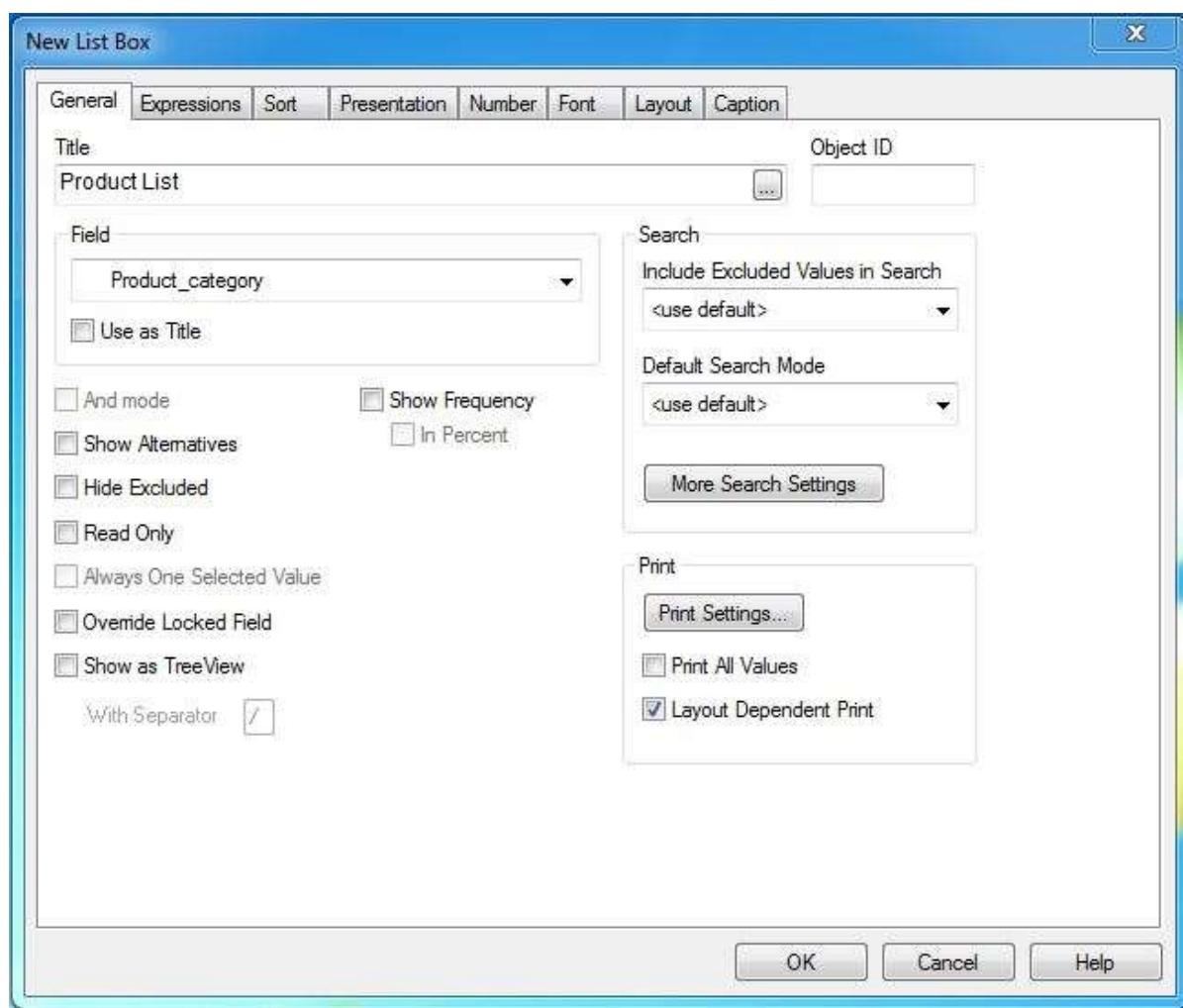
```
LOAD Product_Line,
    Product_category,
    Value
FROM
[C:\Qlikview\data\product_sales.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);
```

## Creating List Box

Creation of **List Box** involves navigating through menu **Layout -> New Sheet Object -> List Box**. The following screen shows these steps.

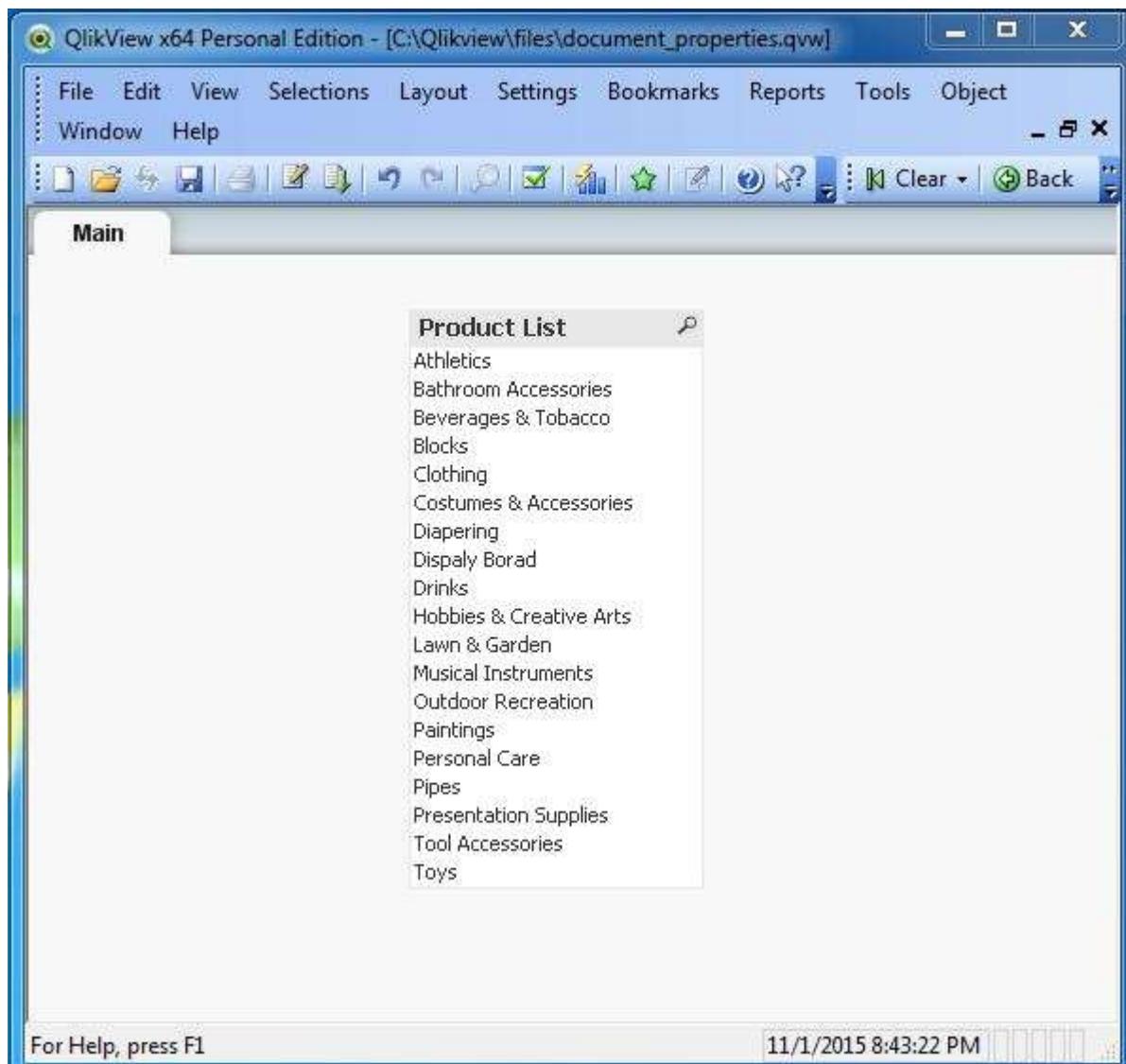


Next, we choose Product category as the field on which we build the list box.



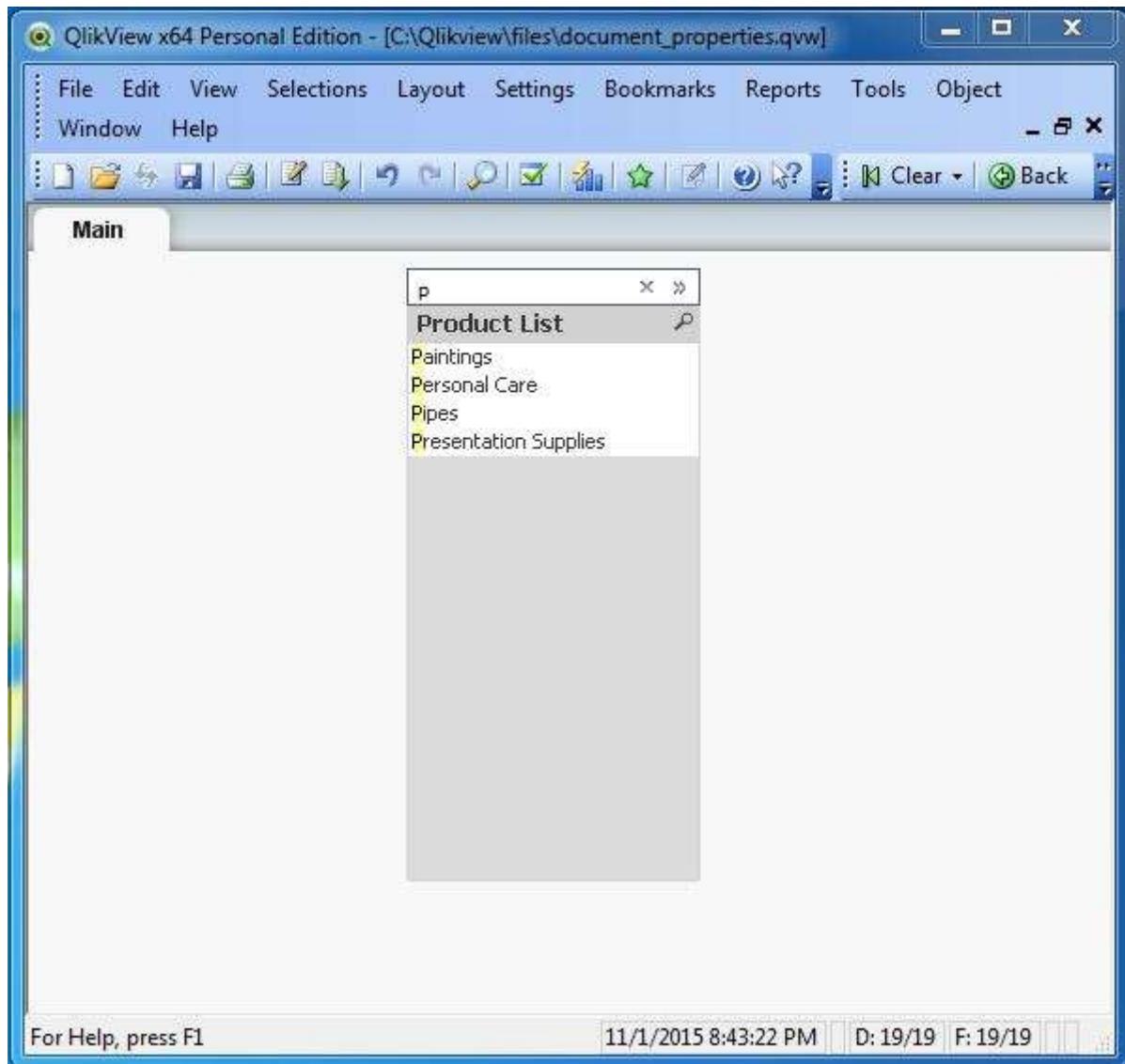
## List Box Data

Finishing the above steps brings the following screen, which shows the values of Product category as a list box.



## Search Option in List Box

When the List Box contains very large number of values, it is difficult to scroll down and look for it. So the search box at the top of the list box can be used to type the search string. The relevant values appear as soon as the first letter is typed.



## List Box and Association

Other Sheet Objects automatically get associated with the List Box and the association is easily observed by selecting values from the list box.

The screenshot shows the QlikView x64 Personal Edition interface. On the left, there is a table titled "Product Sales" with three columns: "Product\_Line", "Product\_category", and "Value". The data in the table is:

Product_Line	Product_category	Value
Arts & Entertainment	Hobbies & Creative Arts	5201
Arts & Entertainment	Musical Instruments	1245
Arts & Entertainment	Paintings	8451

To the right of the table is a "Product List" window containing a scrollable list of categories. The list includes:

- Athletics
- Bathroom Accessories
- Beverages & Tobacco
- Blocks
- Clothing
- Costumes & Accessories
- Diapering
- Dispaly Borad
- Drinks
- Hobbies & Creative Arts
- Lawn & Garden
- Musical Instruments
- Outdoor Recreation
- Paintings
- Personal Care
- Pipes
- Presentation Supplies
- Tool Accessories
- Toys

The "Hobbies & Creative Arts" category is highlighted in the list. At the bottom of the screen, there is a status bar with the text "For Help, press F1", the date and time "11/1/2015 8:43:22 PM\*", and a "3 X 3" icon.

## 26. QlikView – Multi Box

A **Multi Box** represents the list of all the values from multiple fields as drop down values. Similar to list box, the selection of a value in Multi Box highlights the related values in other sheet objects. This helps in faster visual analysis. It is also very useful to follow a drill down path among various sheet objects.

### Input Data

Let us consider the following input data, which represents the sales figure of different product lines and product categories.

```
Product_Line,Product_category,Value
Sporting Goods,Outdoor Recreation,5642
Food, Beverages & Tobacco,2514
Apparel & Accessories,Clothing,2365
Apparel & Accessories,Costumes & Accessories,4487
Sporting Goods,Athletics,812
Health & Beauty,Personal Care,6912
Arts & Entertainment,Hobbies & Creative Arts,5201
Arts & Entertainment,Paintings,8451
Arts & Entertainment,Musical Instruments,1245
Hardware,Tool Accessories,456
Home & Garden,Bathroom Accessories,241
Food,Drinks,1247
Home & Garden,Lawn & Garden,5462
Office Supplies,Presentation Supplies,577
Hardware,Blocks,548
Baby & Toddler,Diapering,1247
Baby & Toddler,Toys,257
Home & Garden,Pipes,1241
Office Supplies,Dispaly Borad,2177
```

### Load Script

The above data is loaded to QlikView memory by using the script editor. Open the Script editor from the File menu or press **Control+E**. Choose the **Table Files** option from the **Data from Files** tab and browse for the file containing the above data. Edit the load script to add the following code. Click **OK** and click **Control+R** to load the data into the QlikView's memory.

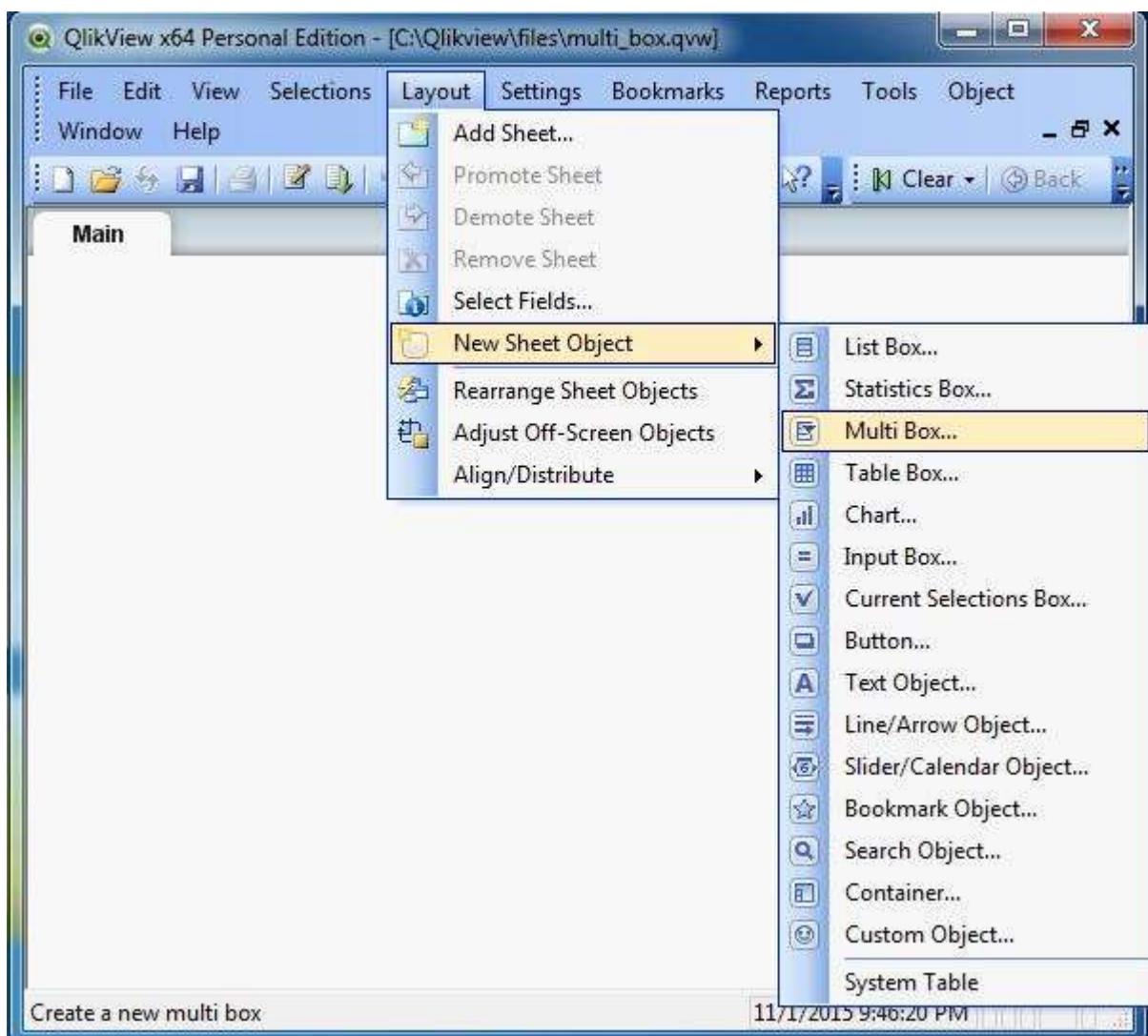
```

LOAD Product_Line,
Product_category,
Value
FROM
[C:\Qlikview\data\product_sales.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);

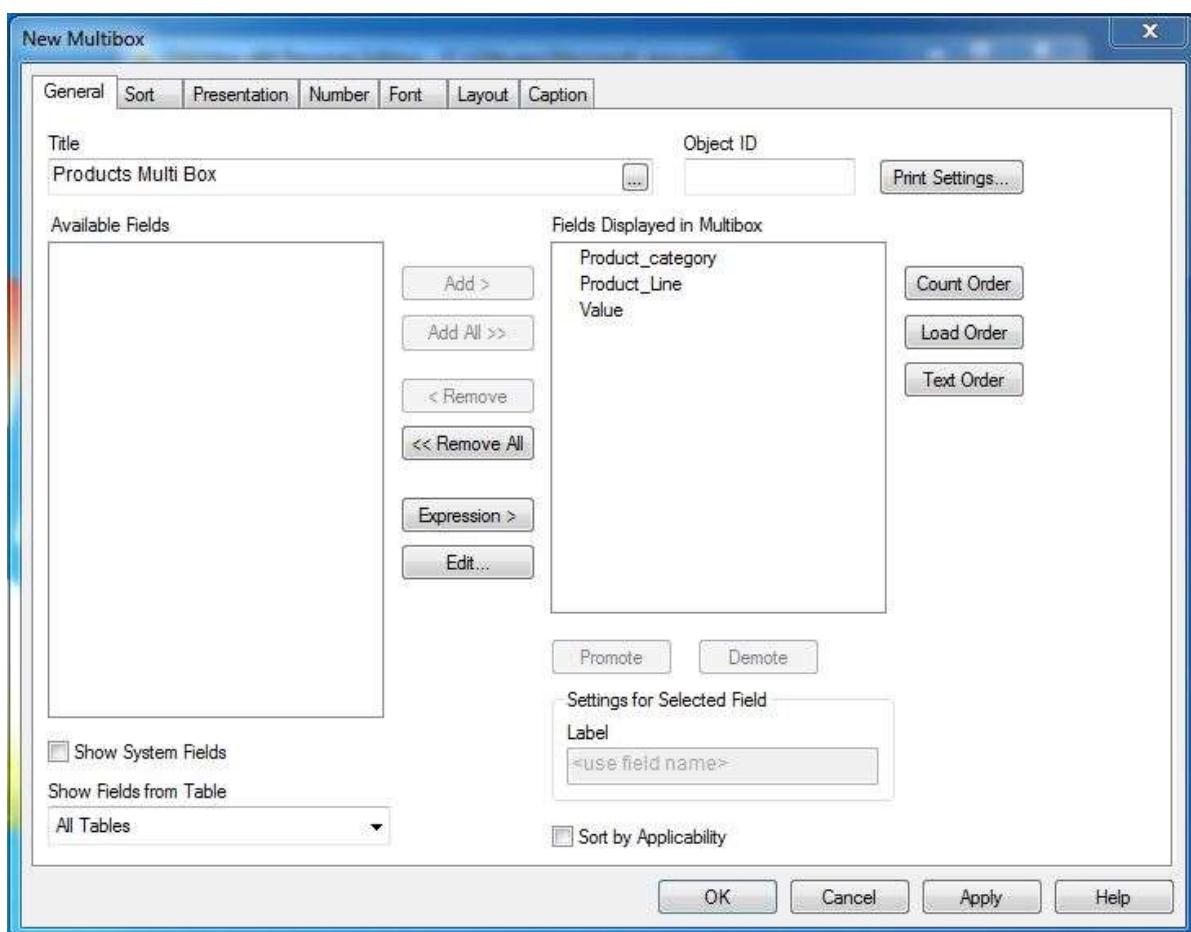
```

## Creating Multi Box

Creation of **Multi Box** involves navigating through menu **Layout -> New Sheet Object -> Multi Box**. The following screen shows these steps.

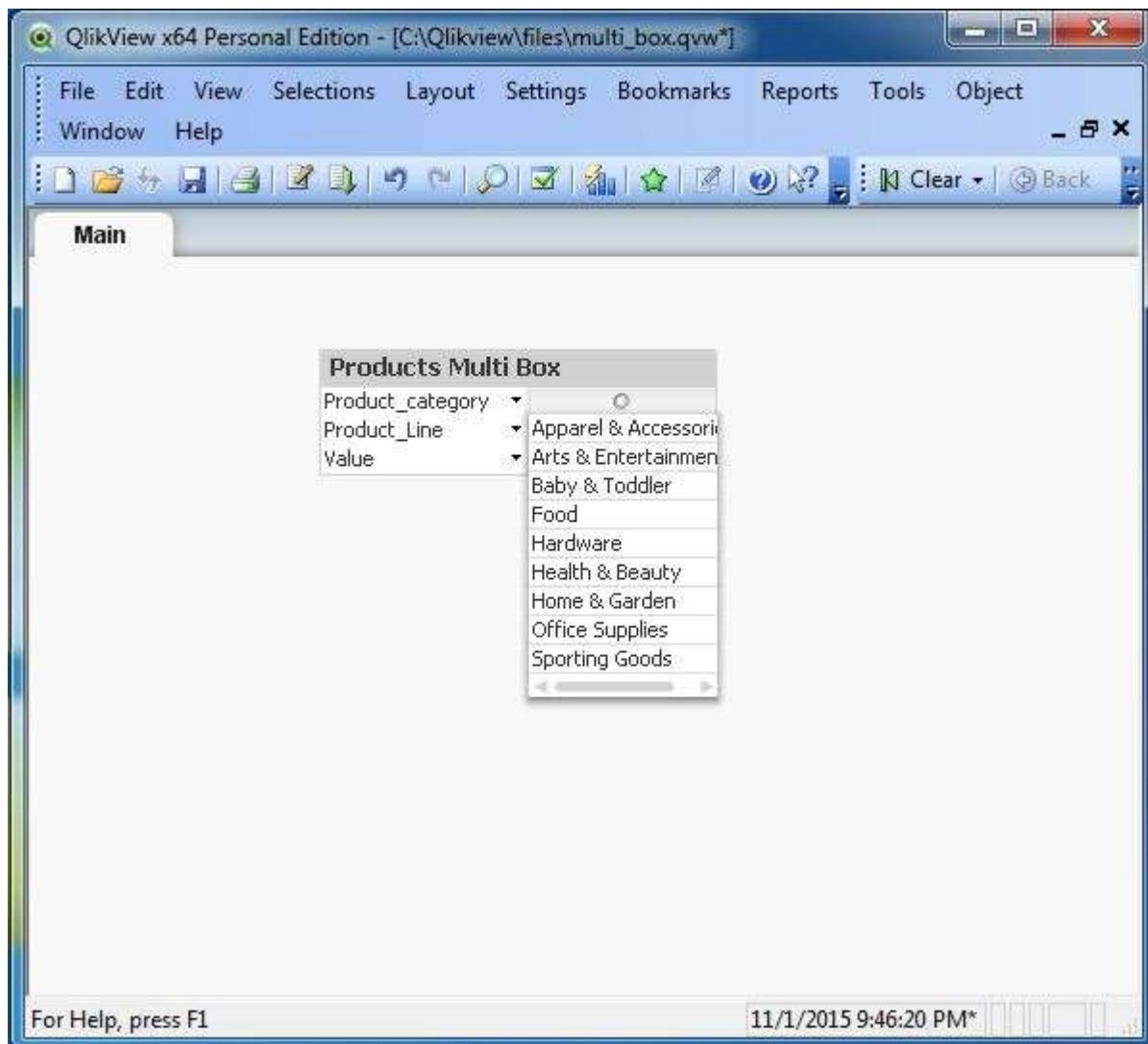


Next we choose the fields of the Products sales tables to build the Multi Box.



## Multi Box Data

Finishing the above steps brings the following screen, which shows the values of Product category as a Multi box.



## Multi Box and Association

Other Sheet Objects automatically get associated with the Multi Box and the association is easily observed by selecting values from the Multi Box.

The screenshot shows the QlikView interface with a 'Main' sheet active. On the left, there is a 'Product List' table with three columns: 'Product\_category', 'Product\_Line', and 'Value'. The data in the table is:

Product_category	Product_Line	Value
Hobbies & Creative Arts	Arts & Entertainment	5201
Musical Instruments	Arts & Entertainment	1245
Paintings	Arts & Entertainment	8451

To the right of the table is a 'Products Multi Box' object. This is a dropdown menu with three items: 'Product\_category', 'Product\_Line', and 'Value'. The 'Product\_Line' item is currently selected, as indicated by a green highlight around its dropdown arrow and the text 'Arts & Entertai...'.

At the bottom of the screen, there is a status bar with the text 'For Help, press F1' on the left and the date/time '11/1/2015 9:46:20 PM' on the right.

## 27. QlikView – Text Object

QlikView **text Object** is used to show some descriptive information about the QlikView report being displayed. It can also show calculations based on certain expressions. It is mainly used for displaying nicely formatted information using colors and different font types in a box separately from the other Sheet Objects.

### Input Data

Let us consider the following input data, which represents the sales figure of different product lines and product categories.

```
Product_Line,Product_category,Value
Sporting Goods,Outdoor Recreation,5642
Food, Beverages & Tobacco,2514
Apparel & Accessories,Clothing,2365
Apparel & Accessories,Costumes & Accessories,4487
Sporting Goods,Athletics,812
Health & Beauty,Personal Care,6912
Arts & Entertainment,Hobbies & Creative Arts,5201
Arts & Entertainment,Paintings,8451
Arts & Entertainment,Musical Instruments,1245
Hardware,Tool Accessories,456
Home & Garden,Bathroom Accessories,241
Food,Drinks,1247
Home & Garden,Lawn & Garden,5462
Office Supplies,Presentation Supplies,577
Hardware,Blocks,548
Baby & Toddler,Diapering,1247
Baby & Toddler,Toys,257
Home & Garden,Pipes,1241
Office Supplies,Dispaly Borad,2177
```

### Load Script

The above data is loaded to QlikView memory by using the script editor. Open the Script editor from the File menu or press **Control+E**. Choose the "Table Files" option from the "Data from Files" tab and browse for the file containing the above data. Edit the load script to add the following code. Click "OK" and press "Control+R" to load the data into the QlikView's memory.

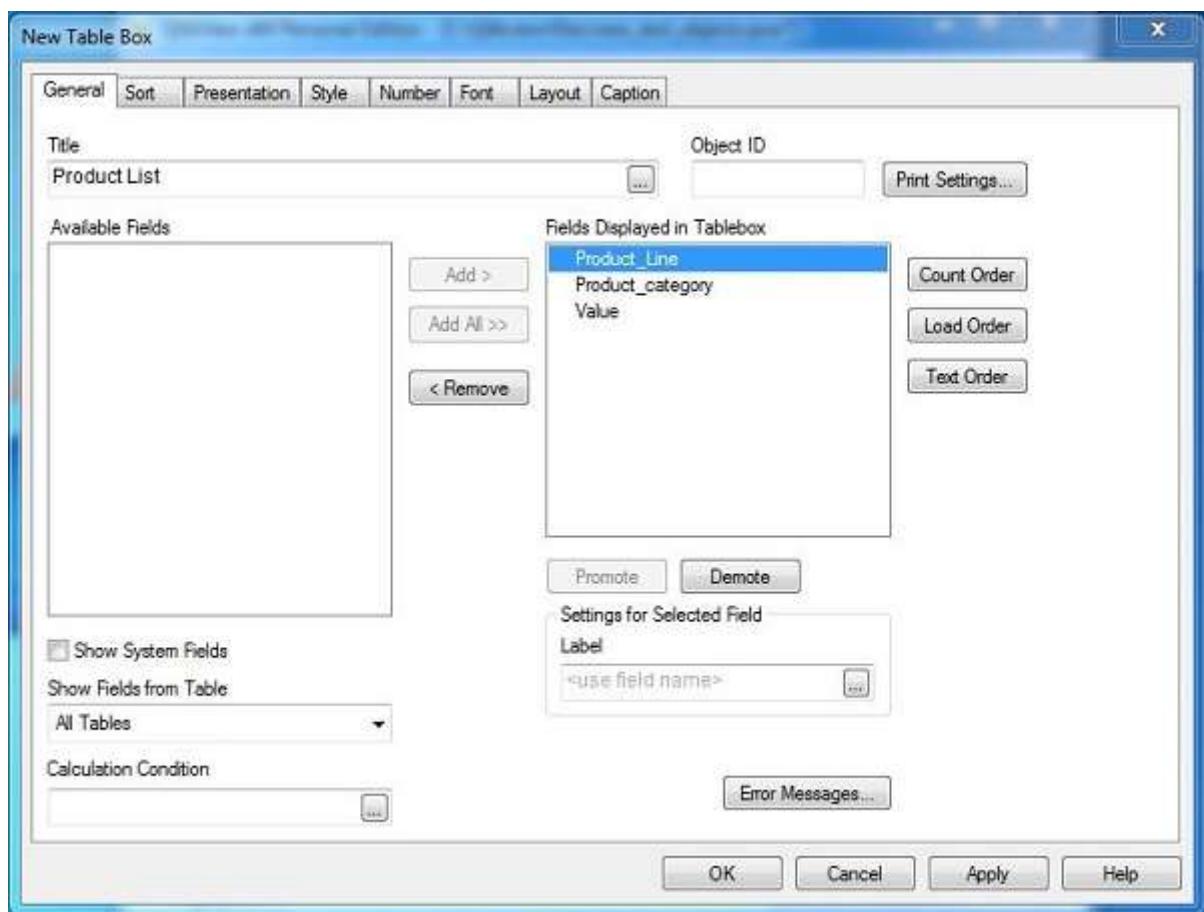
```

LOAD Product_Line,
Product_category,
Value
FROM
[C:\Qlikview\data\product_sales.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);

```

## Creating Table Box

For the above data, let us create a **Table Box**, which will show the data in a tabular form. Go to the menu **Layout -> New Sheet Object -> Table Box** and choose the column as shown below.



Click **Apply** and then **OK** to finish creating the Table box. The following screen appears.

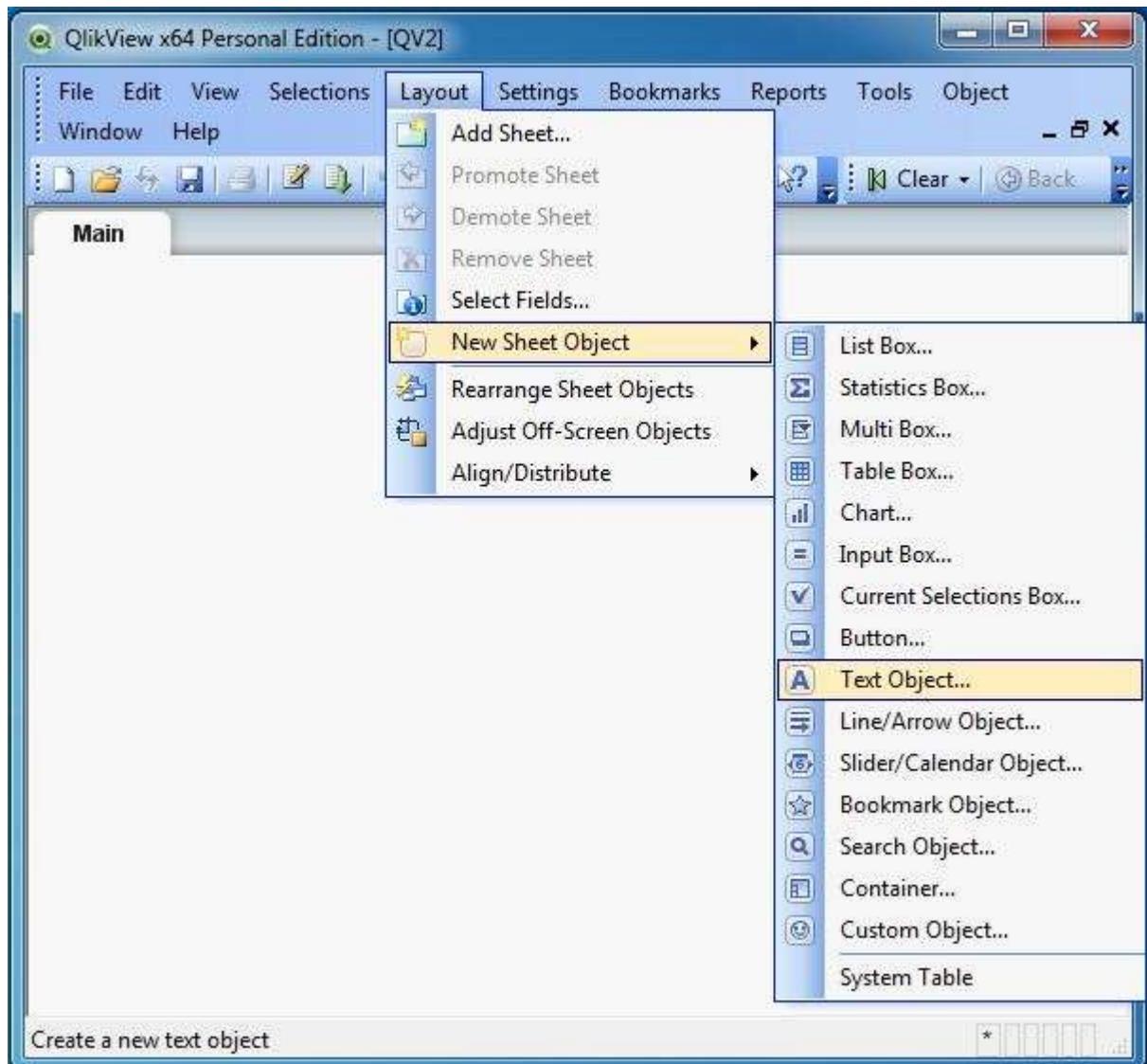
The screenshot shows the QlikView x64 Personal Edition interface. The window title is "QlikView x64 Personal Edition - [C:\Qlikview\files\new\_text\_objects.qvw]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and zoom. The main area is titled "Main" and displays a table titled "Product List". The table has three columns: "Product\_Line", "Product\_category", and "Value". The data in the table is as follows:

Product_Line	Product_category	Value
Apparel & Accessories	Clothing	2365
Apparel & Accessories	Costumes & Accessories	4487
Arts & Entertainment	Hobbies & Creative Arts	5201
Arts & Entertainment	Musical Instruments	1245
Arts & Entertainment	Paintings	8451
Baby & Toddler	Diapering	1247
Baby & Toddler	Toys	257
Food	Beverages & Tobacco	2514
Food	Drinks	1247
Hardware	Blocks	548
Hardware	Tool Accessories	456
Health & Beauty	Personal Care	6912
Home & Garden	Bathroom Accessories	241
Home & Garden	Lawn & Garden	5462
Home & Garden	Pipes	1241
Office Supplies	Dispaly Borad	2177
Office Supplies	Presentation Supplies	577
Sporting Goods	Athletics	812
Sporting Goods	Outdoor Recreation	5642

At the bottom left, it says "For Help, press F1" and at the bottom right, the date and time are "11/2/2015 9:09:32 AM".

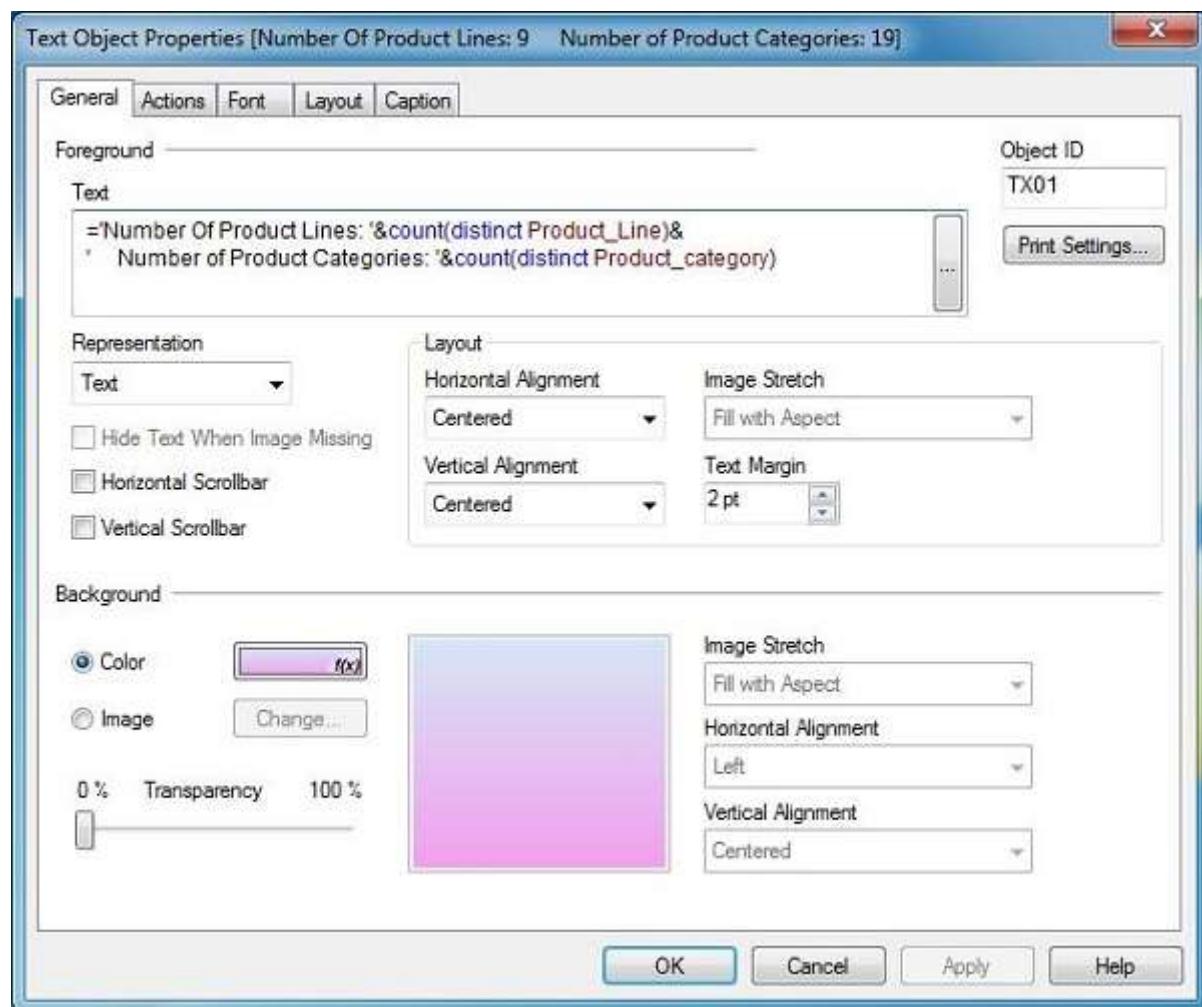
## Creating Text Object

For the above data, let us create a **Text Object**. Go to the menu **Layout -> New Sheet Object -> Text Object** as shown below.



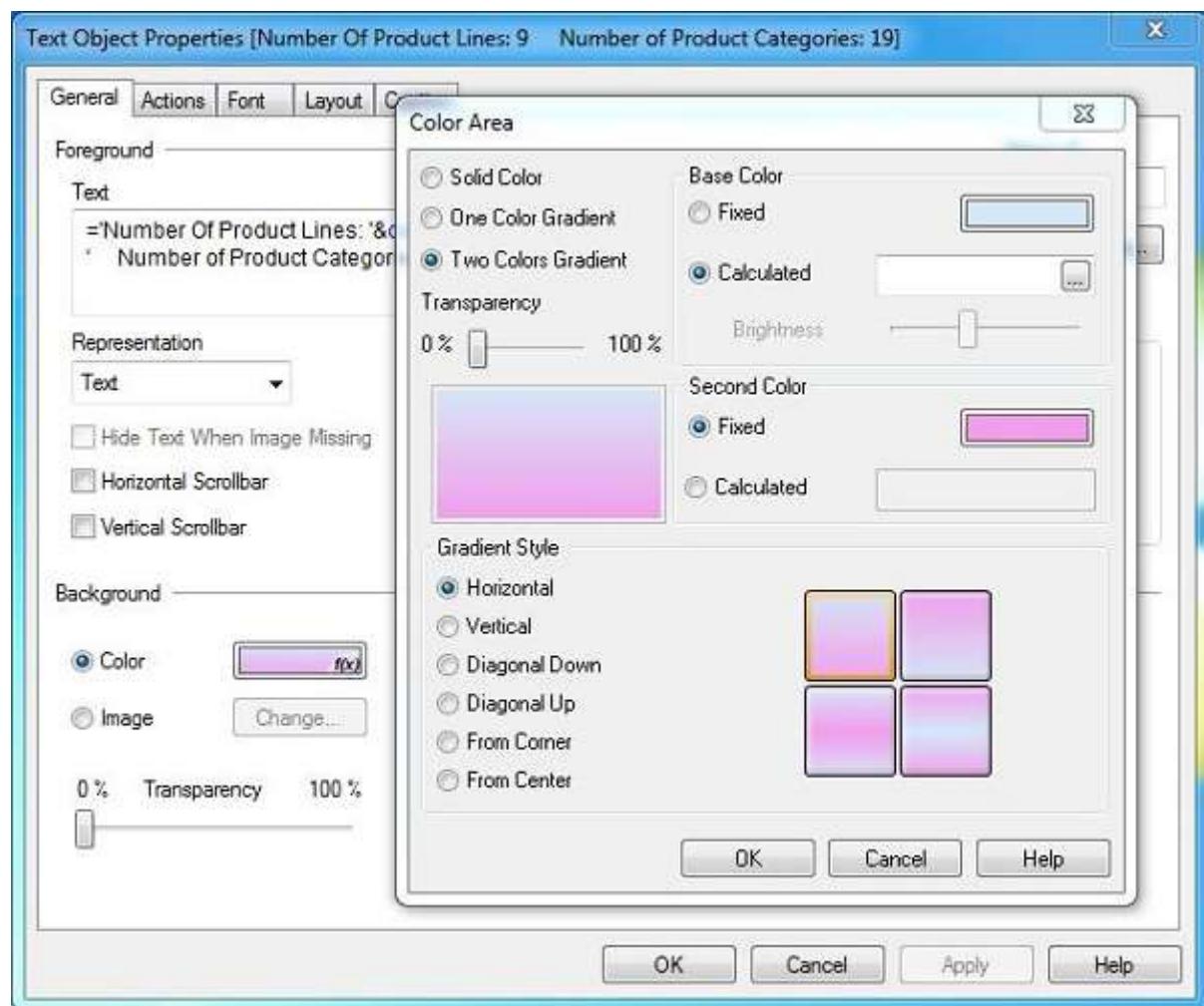
## Creating Text Object Expression

On the text box created above, right click and choose properties. Then enter the content to be displayed on the Text Object in the Text box under the General tab as shown below.



## Choosing the Background Colour

The background color of the Text Object can be set using the background option in the General tab.



## The Text Object

The final Text Object is shown below. If we click on some Product Line to filter it, then the content in the Text Object changes accordingly to reflect the new values.

The screenshot shows the QlikView interface with a 'Main' dashboard. On the left, there is a table titled 'Product List' with columns: Product\_Line, Product\_category, and Value. The table contains 25 rows of product data. On the right, there is a pink rectangular Text Object containing the following text:

**Number Of Product Lines: 9  
Number of Product Categories: 19**

At the bottom left of the screen, there is a message: 'For Help, press F1'. At the bottom right, the date and time are displayed as '11/2/2015 7:14:41 AM'.

Product_Line	Product_category	Value
Sporting Goods	Athletics	812
Home & Garden	Bathroom Accessories	241
Food	Beverages & Tobacco	2514
Hardware	Blocks	548
Apparel & Accessories	Clothing	2365
Apparel & Accessories	Costumes & Accessories	4487
Baby & Toddler	Diapering	1247
Office Supplies	Dispaly Borad	2177
Food	Drinks	1247
Arts & Entertainment	Hobbies & Creative Arts	5201
Home & Garden	Lawn & Garden	5462
Arts & Entertainment	Musical Instruments	1245
Sporting Goods	Outdoor Recreation	5642
Arts & Entertainment	Paintings	8451
Health & Beauty	Personal Care	6912
Home & Garden	Pipes	1241
Office Supplies	Presentation Supplies	577
Hardware	Tool Accessories	456
Baby & Toddler	Toys	257

## 28. QlikView – Bar Chart

Bar charts are very widely used charting method to study the relation between two dimensions in form of bars. The height of the bar in the graph represents the value of one dimension. The number of bars represent the sequence of values or grouped values of another dimension.

### Input Data

Let us consider the following input data, which represents the sales figure of different product lines and product categories.

```
Product_Line,Product_category,Value
Sporting Goods,Outdoor Recreation,5642
Food, Beverages & Tobacco,2514
Apparel & Accessories,Clothing,2365
Apparel & Accessories,Costumes & Accessories,4487
Sporting Goods,Athletics,812
Health & Beauty,Personal Care,6912
Arts & Entertainment,Hobbies & Creative Arts,5201
Arts & Entertainment,Paintings,8451
Arts & Entertainment,Musical Instruments,1245
Hardware,Tool Accessories,456
Home & Garden,Bathroom Accessories,241
Food,Drinks,1247
Home & Garden,Lawn & Garden,5462
Office Supplies,Presentation Supplies,577
Hardware,Blocks,548
Baby & Toddler,Diapering,1247
Baby & Toddler,Toys,257
Home & Garden,Pipes,1241
Office Supplies,Dispaly Borad,2177
```

### Load Script

The above data is loaded to the QlikView memory by using the script editor. Open the Script editor from the File menu or press **Control+E**. Choose the "Table Files" option from the "Data from Files" tab and browse for the file containing the above data. Edit the load script to add the following code. Click "OK" and press "**Control+R**" to load the data into the QlikView's memory.

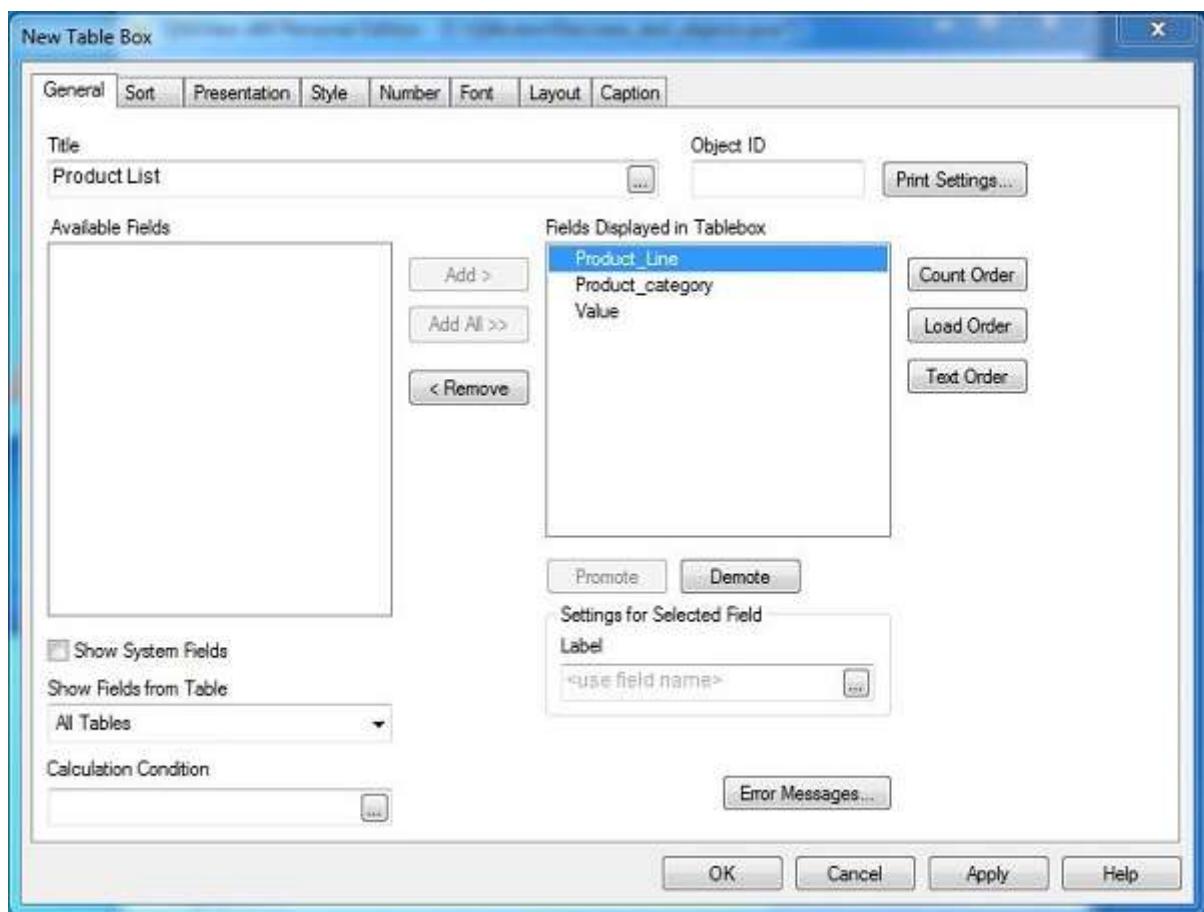
```

LOAD Product_Line,
Product_category,
Value
FROM
[C:\Qlikview\data\product_sales.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);

```

## Creating Table Box (Sheet Object)

For the above data, let us create a **Table Box**, which will show the data in a tabular form. Go to the menu **Layout -> New Sheet Object -> Table Box** and choose the column as shown below.



Click **Apply** and then **OK** to finish creating the Table box. The below given screen appears.

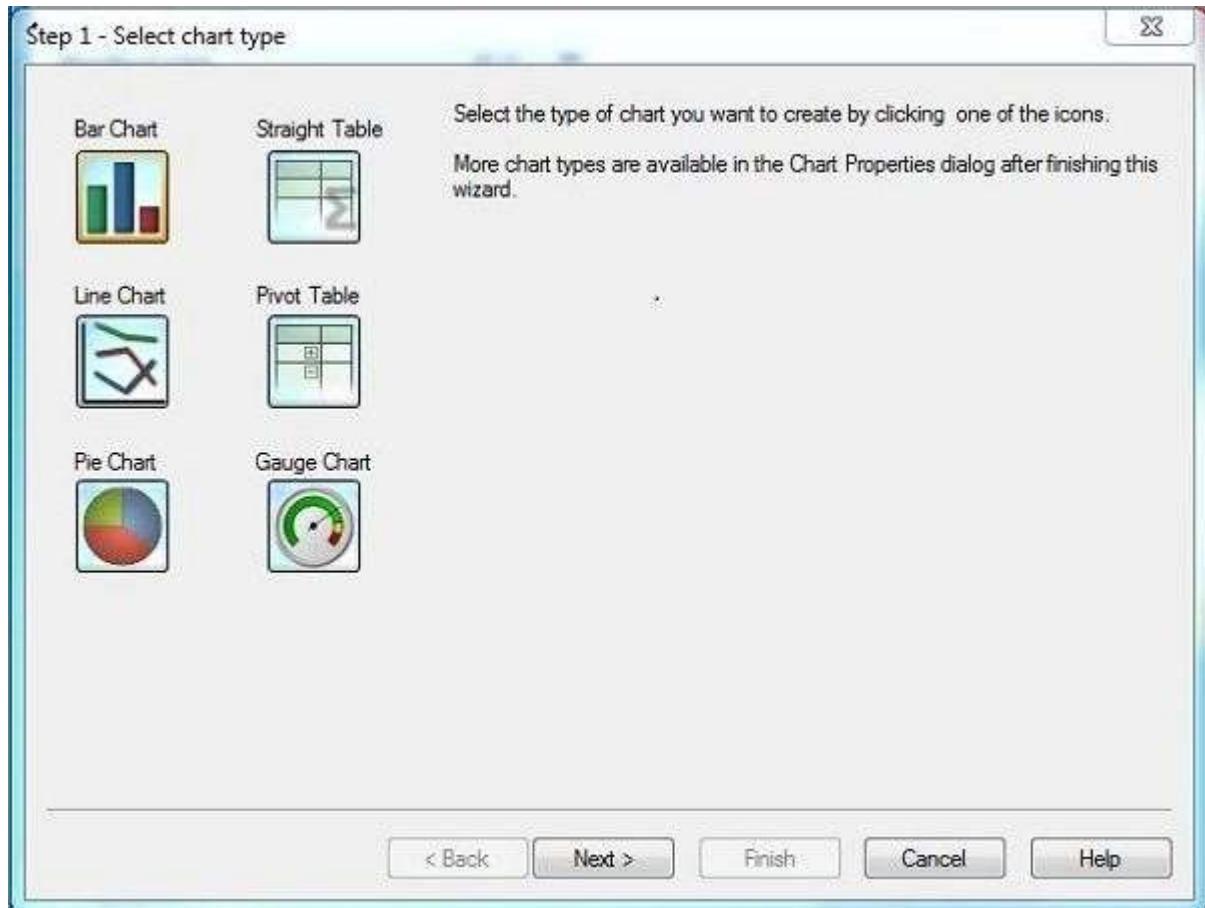
The screenshot shows the QlikView x64 Personal Edition interface. The title bar reads "QlikView x64 Personal Edition - [C:\Qlikview\files\new\_text\_objects.qvw]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and zoom. The main window is titled "Main" and displays a table titled "Product List". The table has three columns: "Product\_Line", "Product\_category", and "Value". The data in the table is as follows:

Product_Line	Product_category	Value
Apparel & Accessories	Clothing	2365
Apparel & Accessories	Costumes & Accessories	4487
Arts & Entertainment	Hobbies & Creative Arts	5201
Arts & Entertainment	Musical Instruments	1245
Arts & Entertainment	Paintings	8451
Baby & Toddler	Diapering	1247
Baby & Toddler	Toys	257
Food	Beverages & Tobacco	2514
Food	Drinks	1247
Hardware	Blocks	548
Hardware	Tool Accessories	456
Health & Beauty	Personal Care	6912
Home & Garden	Bathroom Accessories	241
Home & Garden	Lawn & Garden	5462
Home & Garden	Pipes	1241
Office Supplies	Dispaly Borad	2177
Office Supplies	Presentation Supplies	577
Sporting Goods	Athletics	812
Sporting Goods	Outdoor Recreation	5642

At the bottom left, it says "For Help, press F1" and at the bottom right, the date and time are "11/2/2015 9:09:32 AM".

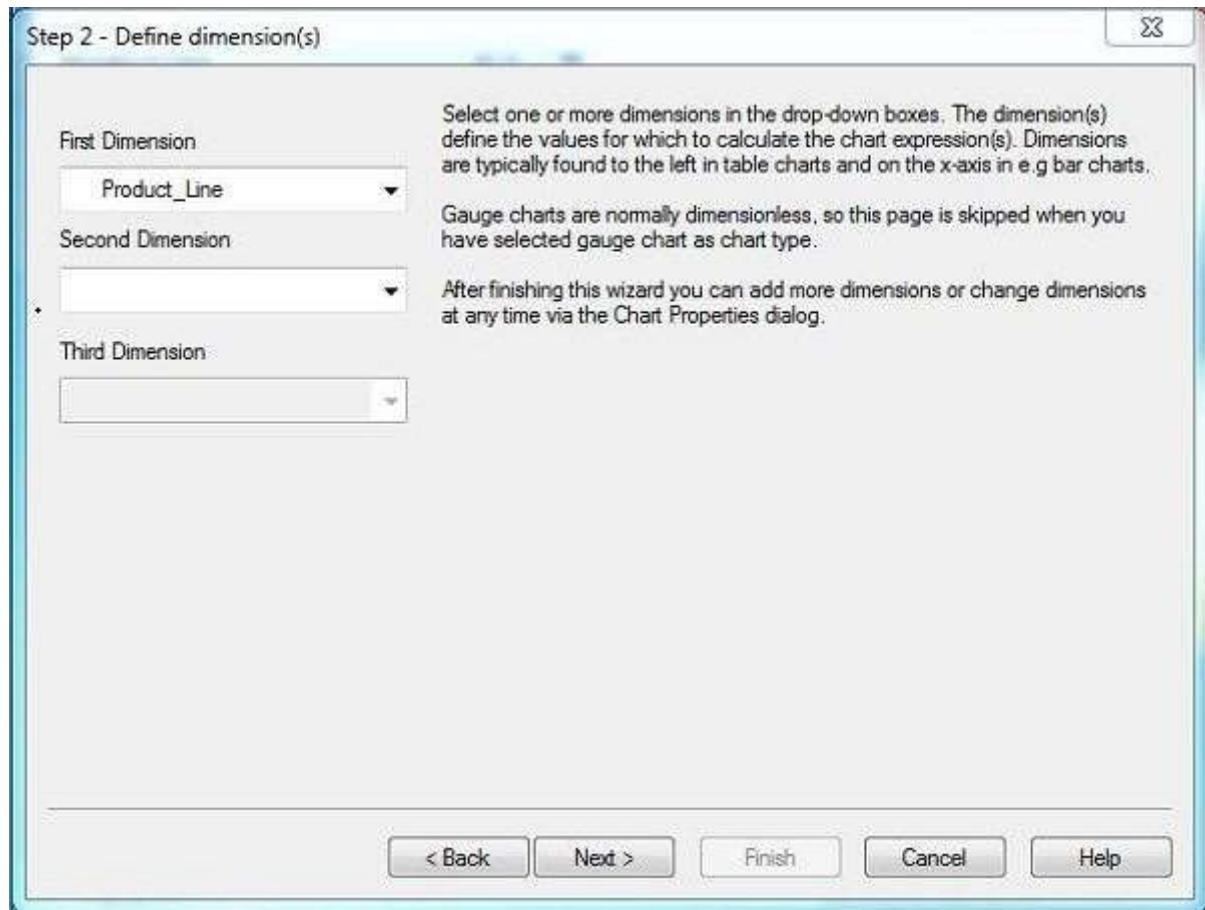
## Using the Quick Chart Wizard

To start creating a bar chart, we will use the quick chart wizard. On clicking it, the following screen appears which prompts for selecting the chart type. Choose **bar Chart** and click Next.



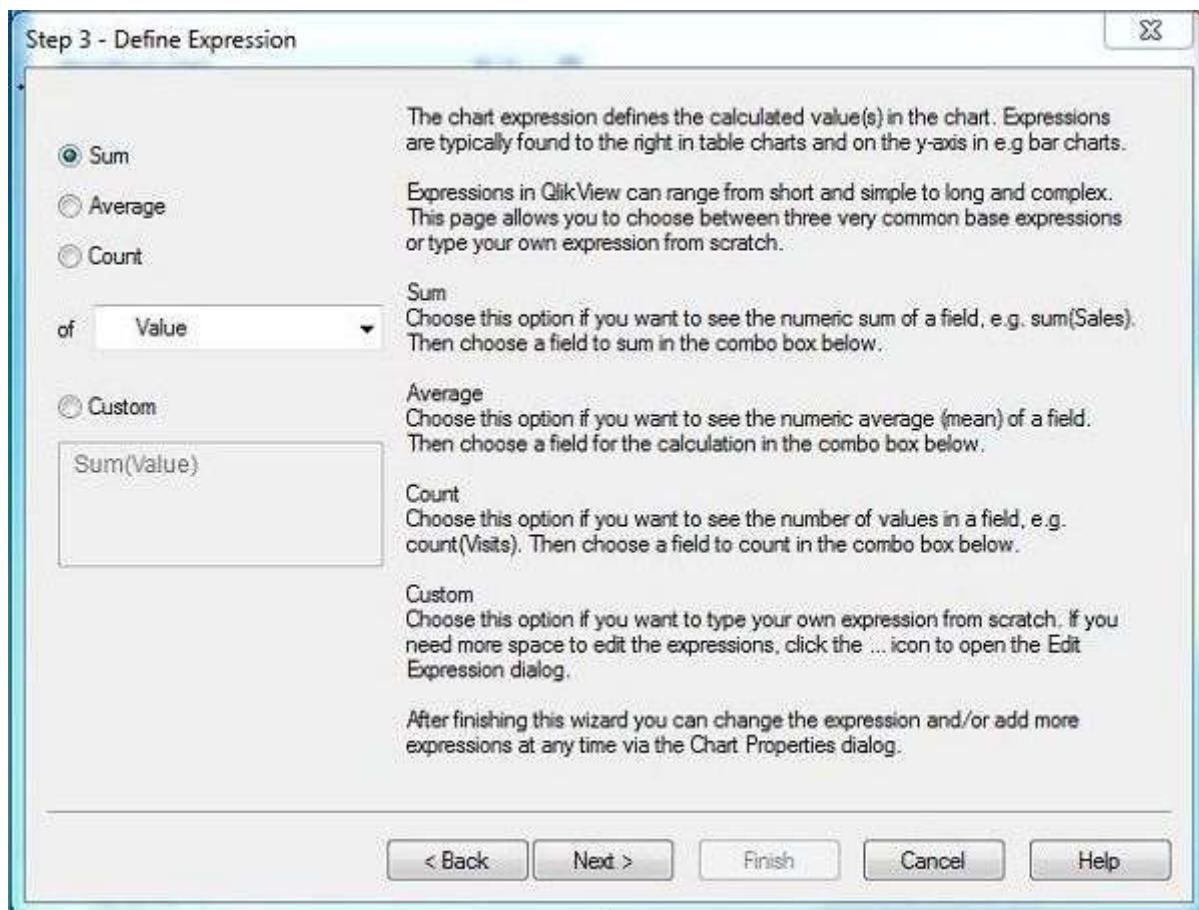
## Choose the Chart Dimension

Choose Product Line as the First Dimension.



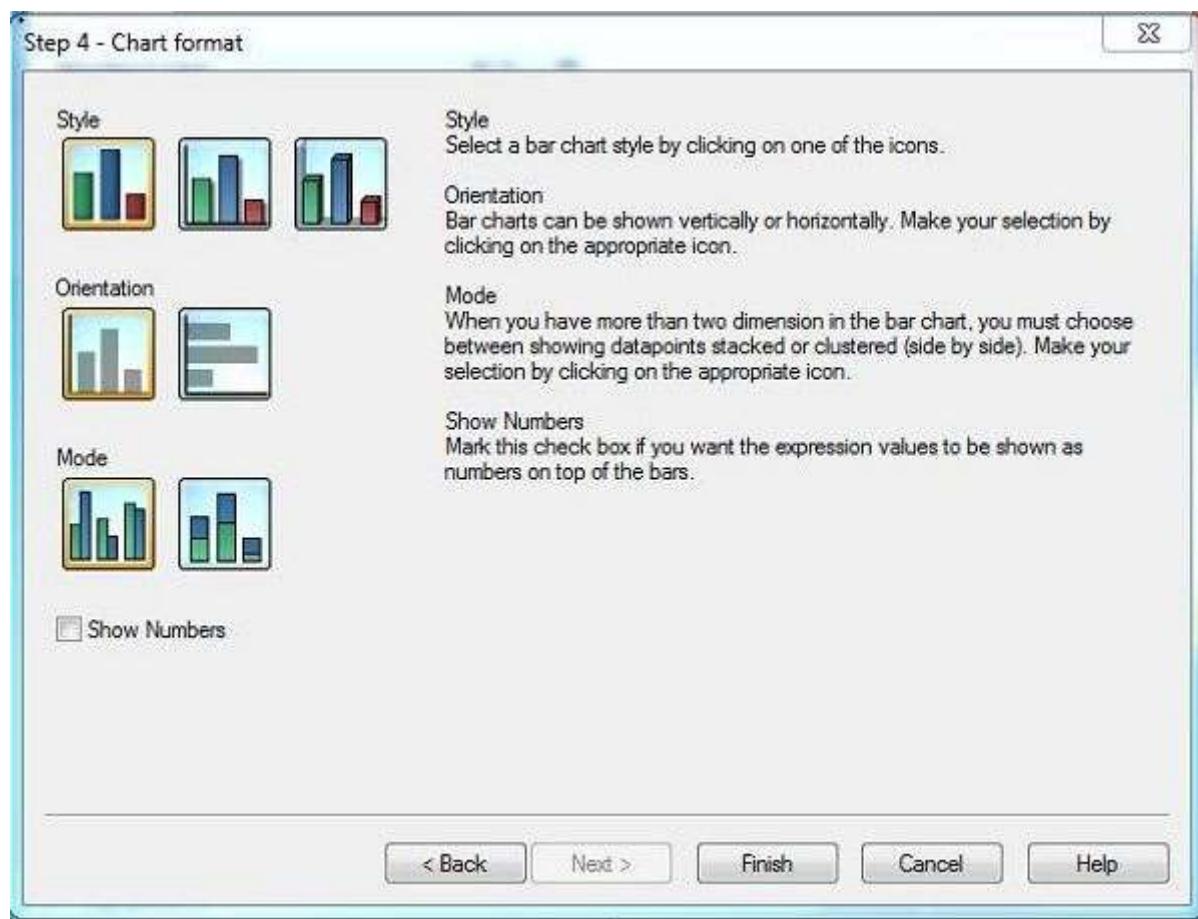
## Choose the Chart Expression

The chart expression is used to apply the functions like **Sum, Average, or Count** on the fields with numeric values. We will apply the Sum function on the field named Value. Click Next.



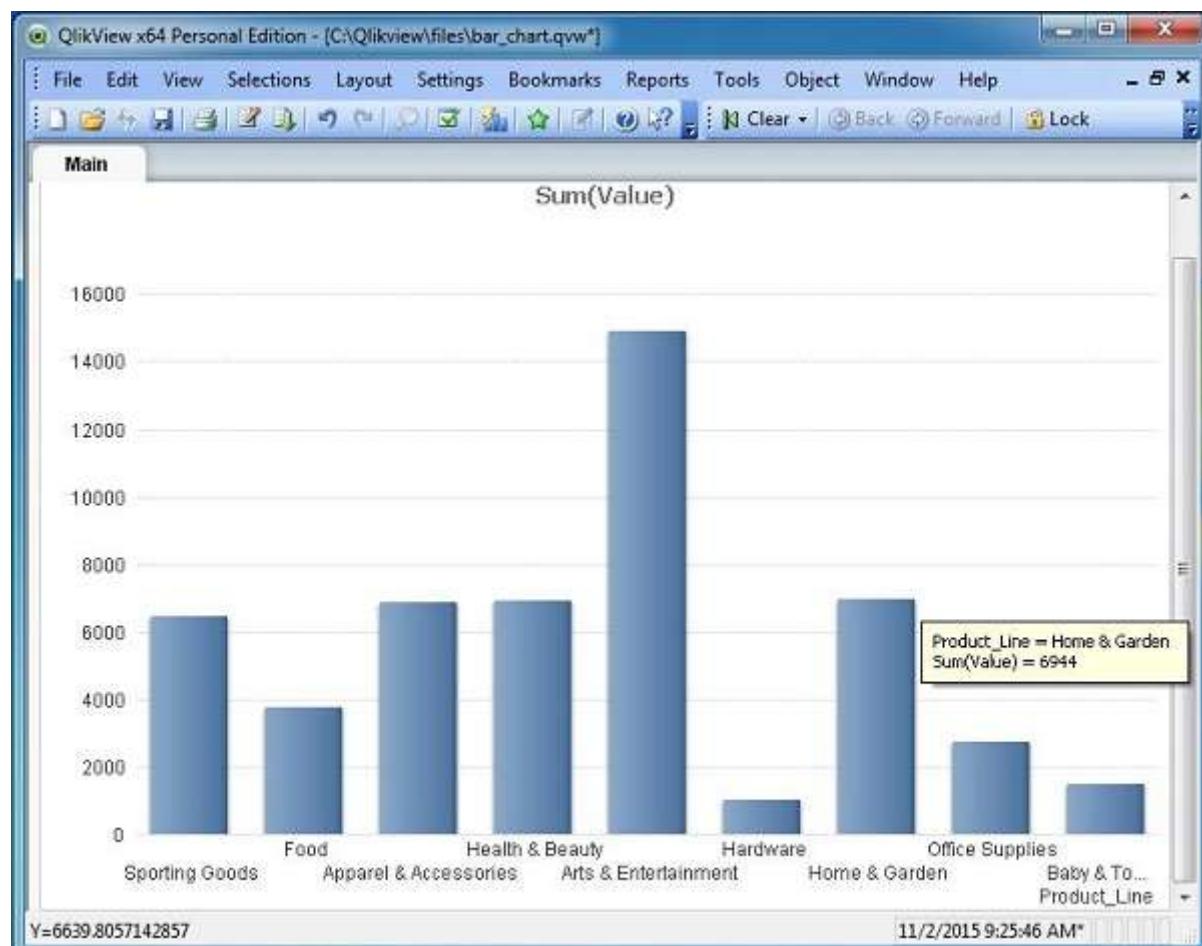
## Choose the Chart Format

The Chart format defines the style and orientation of the chart. We choose the first option in each category. Click Next.



## The Bar Chart

The Bar chart appears as shown below. It shows the height of the field value for different product lines.



## 29. QlikView – Pie Chart

A **pie-chart** is a representation of values as slices of a circle with different colors. The slices are labeled and the numbers corresponding to each slice is also represented in the chart. QlikView creates pie-chart using the chart wizard or chart Sheet Object.

### Input Data

Let us consider the following input data, which represents the sales figure of different product lines and product categories.

```
Product_Line,Product_category,Value
Sporting Goods,Outdoor Recreation,5642
Food, Beverages & Tobacco,2514
Apparel & Accessories,Clothing,2365
Apparel & Accessories,Costumes & Accessories,4487
Sporting Goods,Athletics,812
Health & Beauty,Personal Care,6912
Arts & Entertainment,Hobbies & Creative Arts,5201
Arts & Entertainment,Paintings,8451
Arts & Entertainment,Musical Instruments,1245
Hardware,Tool Accessories,456
Home & Garden,Bathroom Accessories,241
Food,Drinks,1247
Home & Garden,Lawn & Garden,5462
Office Supplies,Presentation Supplies,577
Hardware,Blocks,548
Baby & Toddler,Diapering,1247
Baby & Toddler,Toys,257
Home & Garden,Pipes,1241
Office Supplies,Dispaly Borad,2177
```

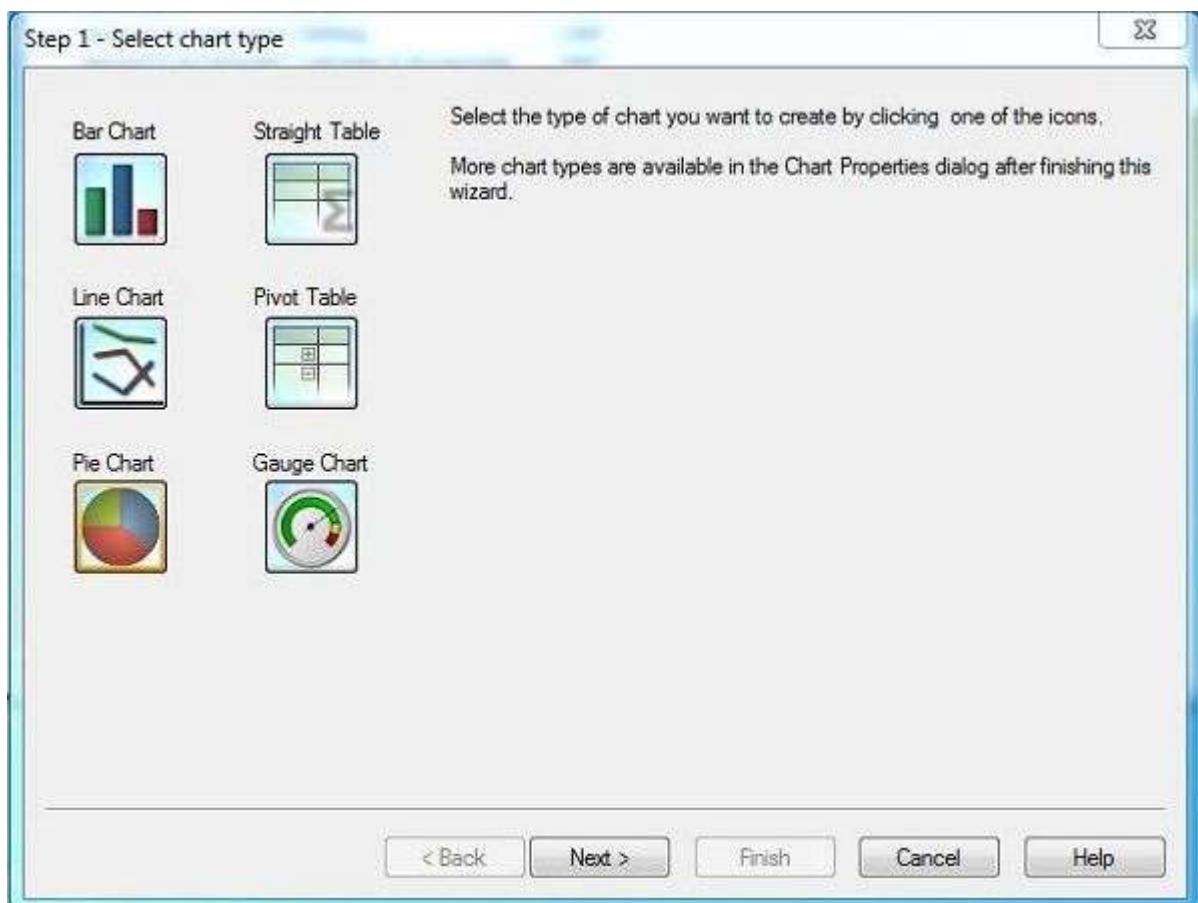
### Load Script

The above data is loaded to QlikView memory by using the script editor. Open the Script editor from the File menu or press **Control+E**. Choose the "Table Files" option from the "Data from Files" tab and browse for the file containing the above data. Edit the load script to add the following code. Click "OK" and press "Control+R" to load the data into the QlikView's memory.

```
LOAD Product_Line,  
    Product_category,  
    Value  
FROM  
[C:\Qlikview\data\product_sales.csv]  
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);
```

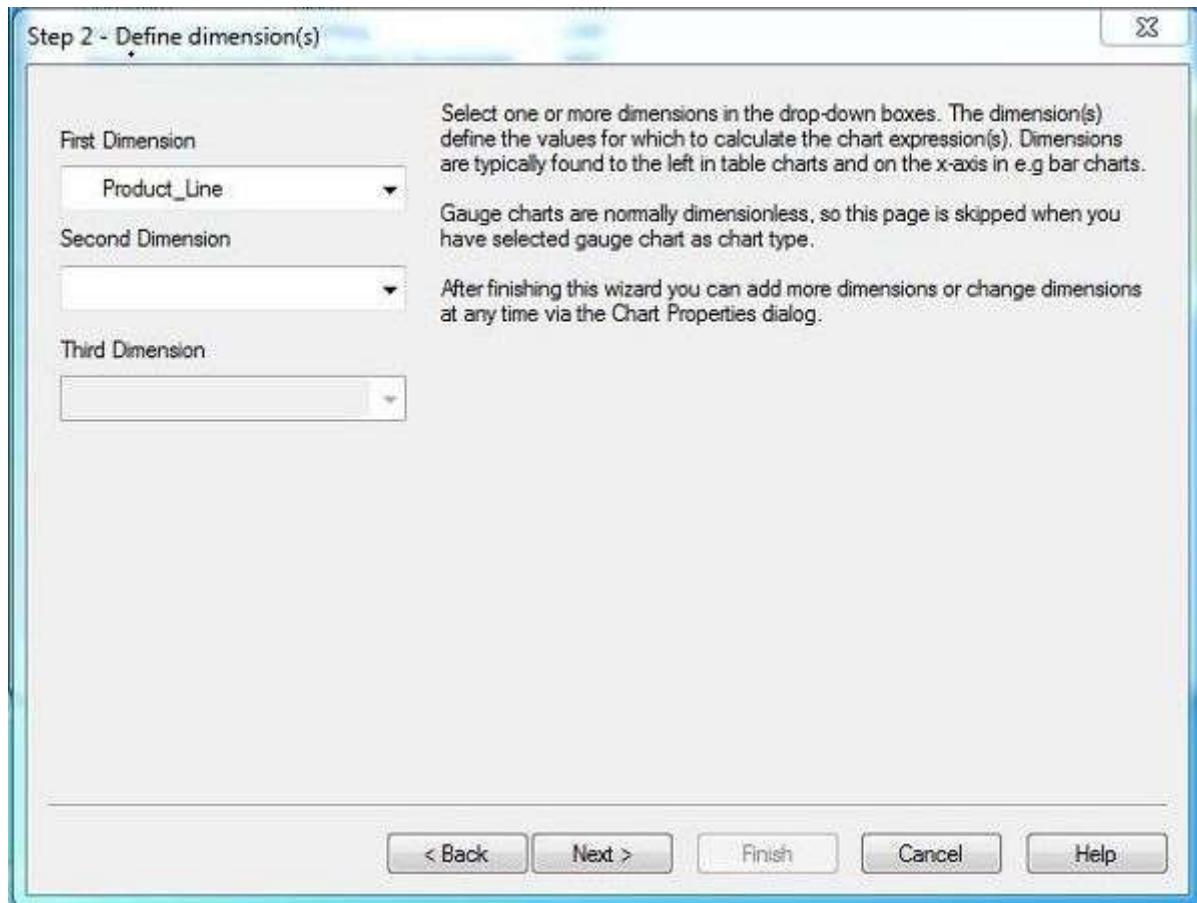
## Using the Quick Chart Wizard

To start creating a Pie chart, we will use the quick chart wizard. On clicking it, the following screen appears which prompts for selecting the chart type. Choose **Pie Chart** and click Next.



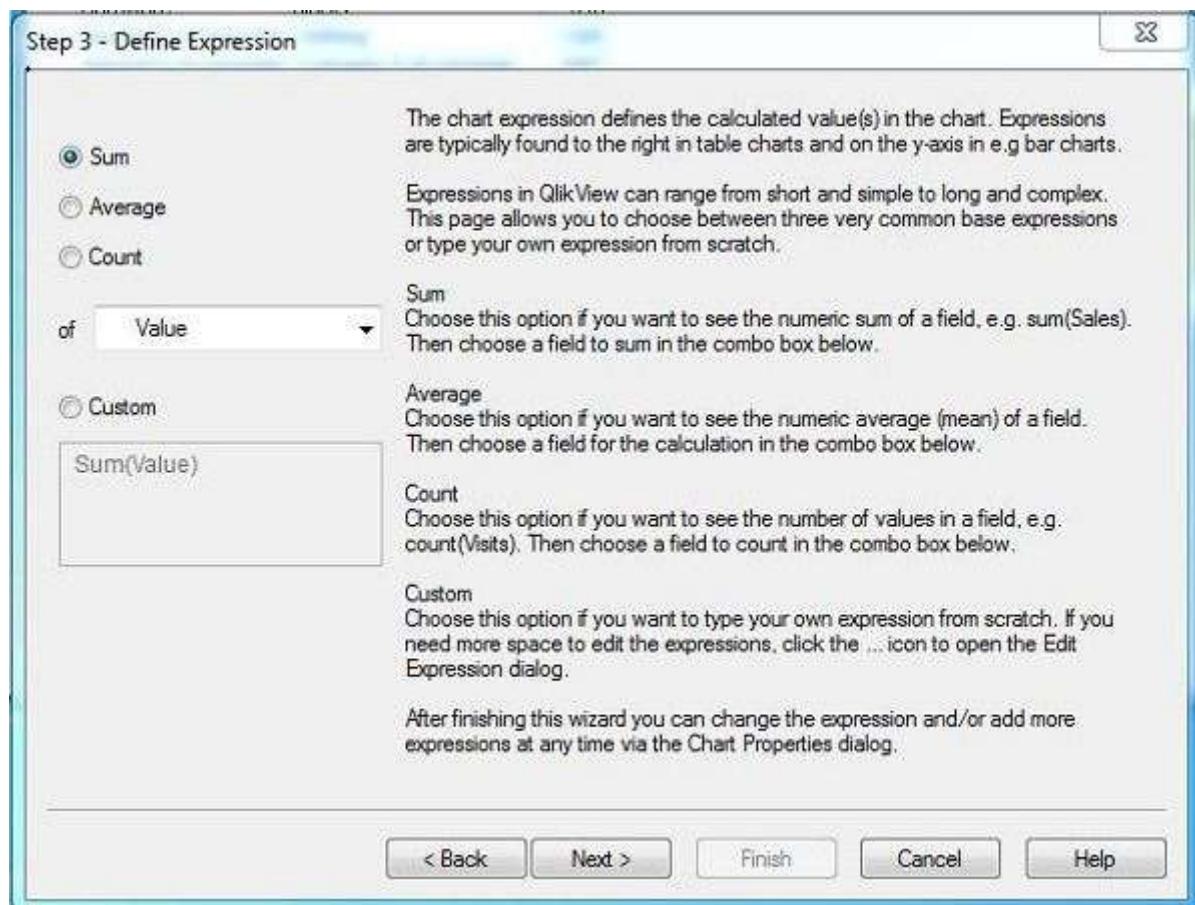
## Choose the Chart Dimension

Choose Product Line as the First Dimension.



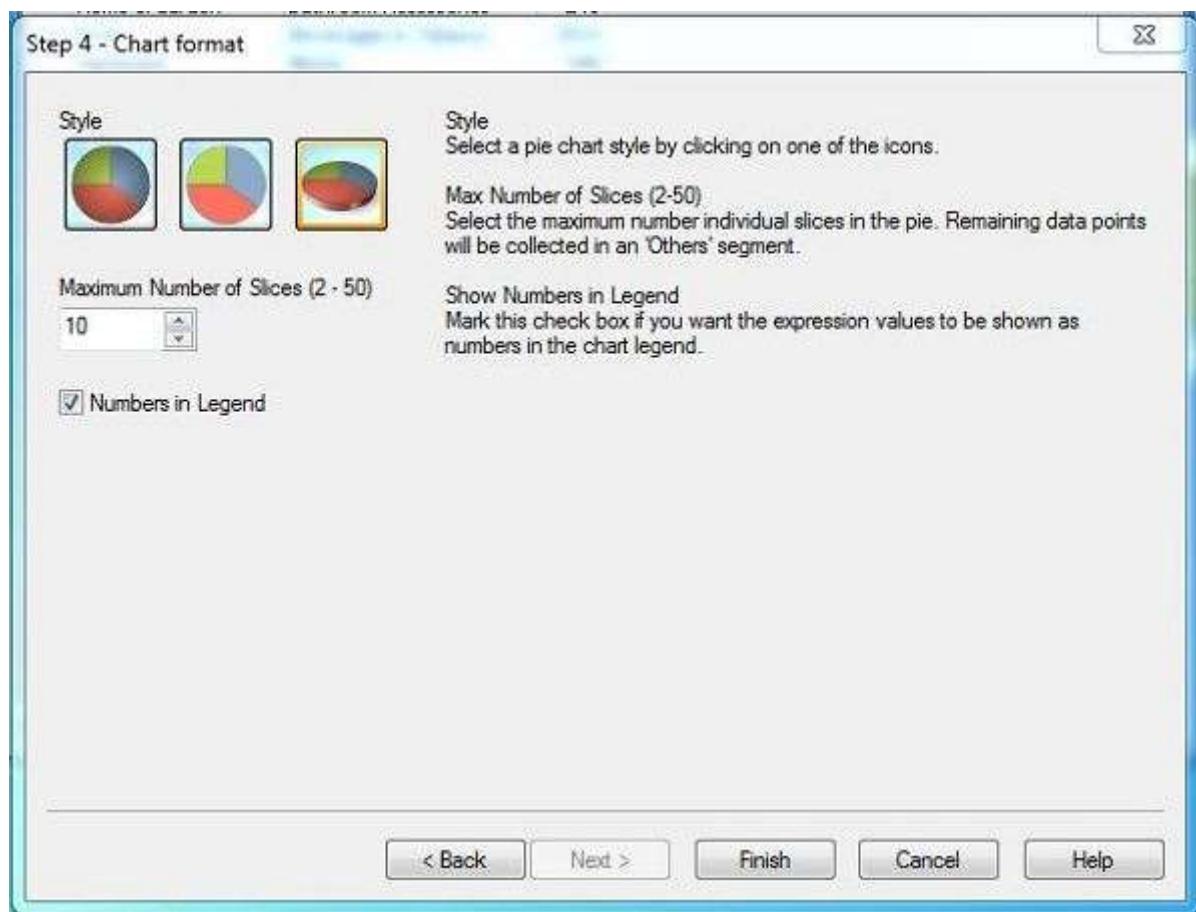
## Choose the Chart Expression

The chart expression is used to apply the functions like **Sum, Average or Count** on the fields with numeric values. We will apply the Sum function on the field named Value. Click Next.



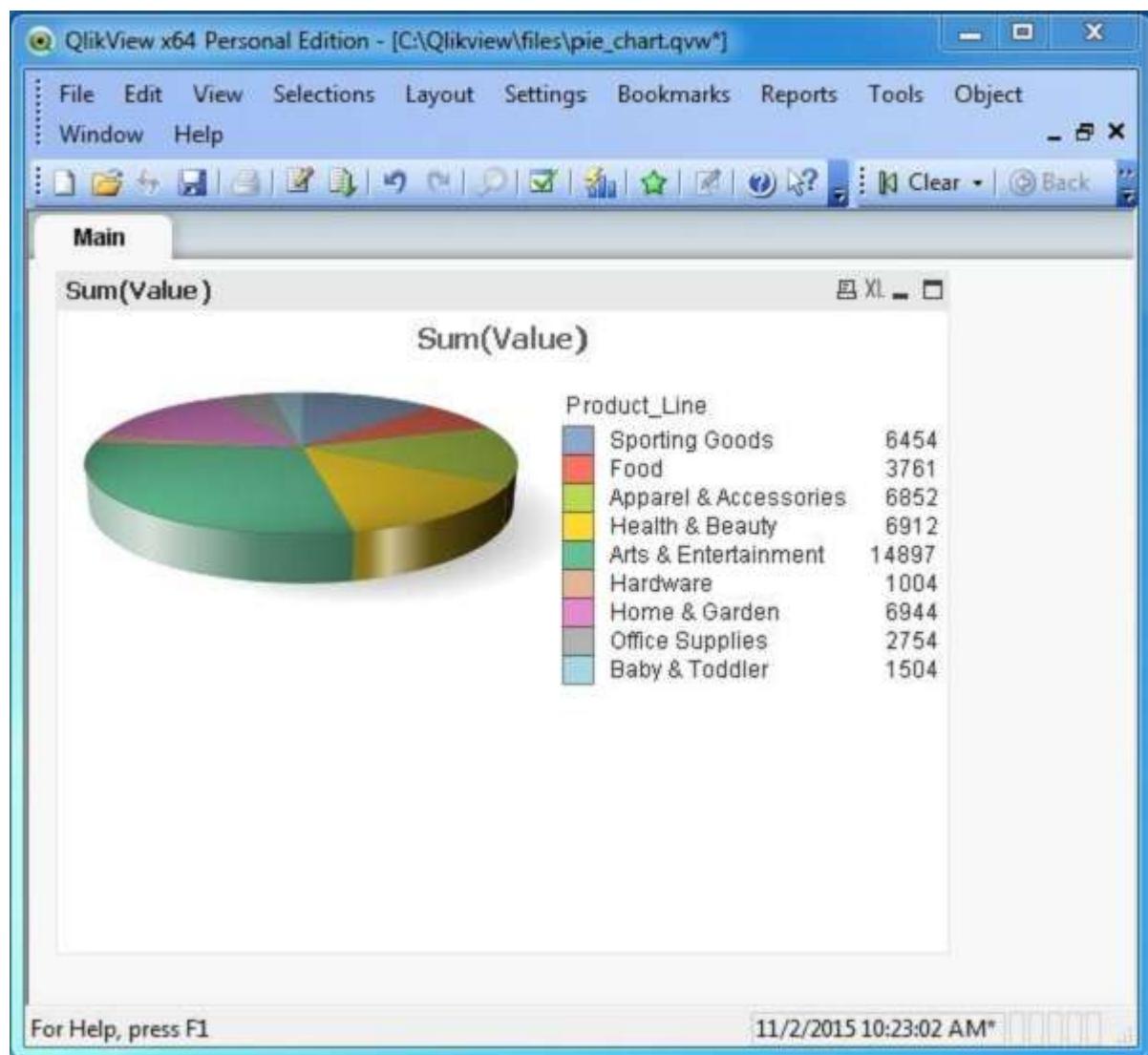
## Choose the Chart Format

The Chart format defines the style and orientation of the chart. We choose the third option. Click Next.



## The Pie Chart

The Bar chart appears as shown below. It shows the height of the field value for different product lines.



## 30. QlikView – Dashboard

A Dashboard is a powerful feature to display values from many fields simultaneously. QlikView's feature of data association in memory can display the dynamic values in all the sheet objects.

### Input Data

Let us consider the following input data, which represents the sales figure of different product lines and product categories.

```
Product_Line,Product_category,Value
Sporting Goods,Outdoor Recreation,5642
Food, Beverages & Tobacco,2514
Apparel & Accessories,Clothing,2365
Apparel & Accessories,Costumes & Accessories,4487
Sporting Goods,Athletics,812
Health & Beauty,Personal Care,6912
Arts & Entertainment,Hobbies & Creative Arts,5201
Arts & Entertainment,Paintings,8451
Arts & Entertainment,Musical Instruments,1245
Hardware,Tool Accessories,456
Home & Garden,Bathroom Accessories,241
Food,Drinks,1247
Home & Garden,Lawn & Garden,5462
Office Supplies,Presentation Supplies,577
Hardware,Blocks,548
Baby & Toddler,Diapering,1247
Baby & Toddler,Toys,257
Home & Garden,Pipes,1241
Office Supplies,Dispaly Borad,2177
```

### Load Script

The above data is loaded to QlikView memory by using the script editor. Open the Script editor from the File menu or press **Control+E**. Choose the "Table Files" option from the "Data from Files" tab and browse for the file containing the above data. Edit the load script to add the following code. Click "OK" and press "Control+R" to load the data into the QlikView's memory.

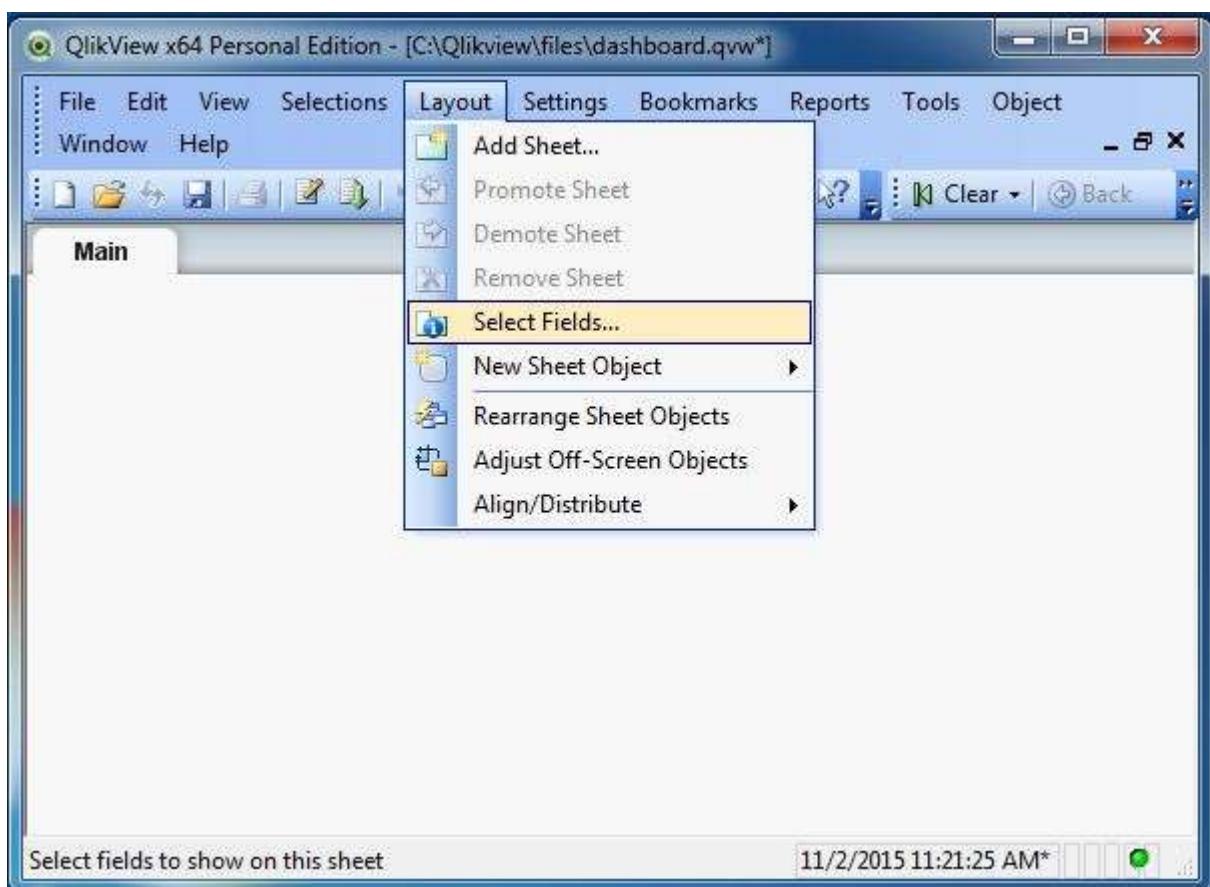
```

LOAD Product_Line,
Product_category,
Value
FROM
[C:\Qlikview\data\product_sales.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);

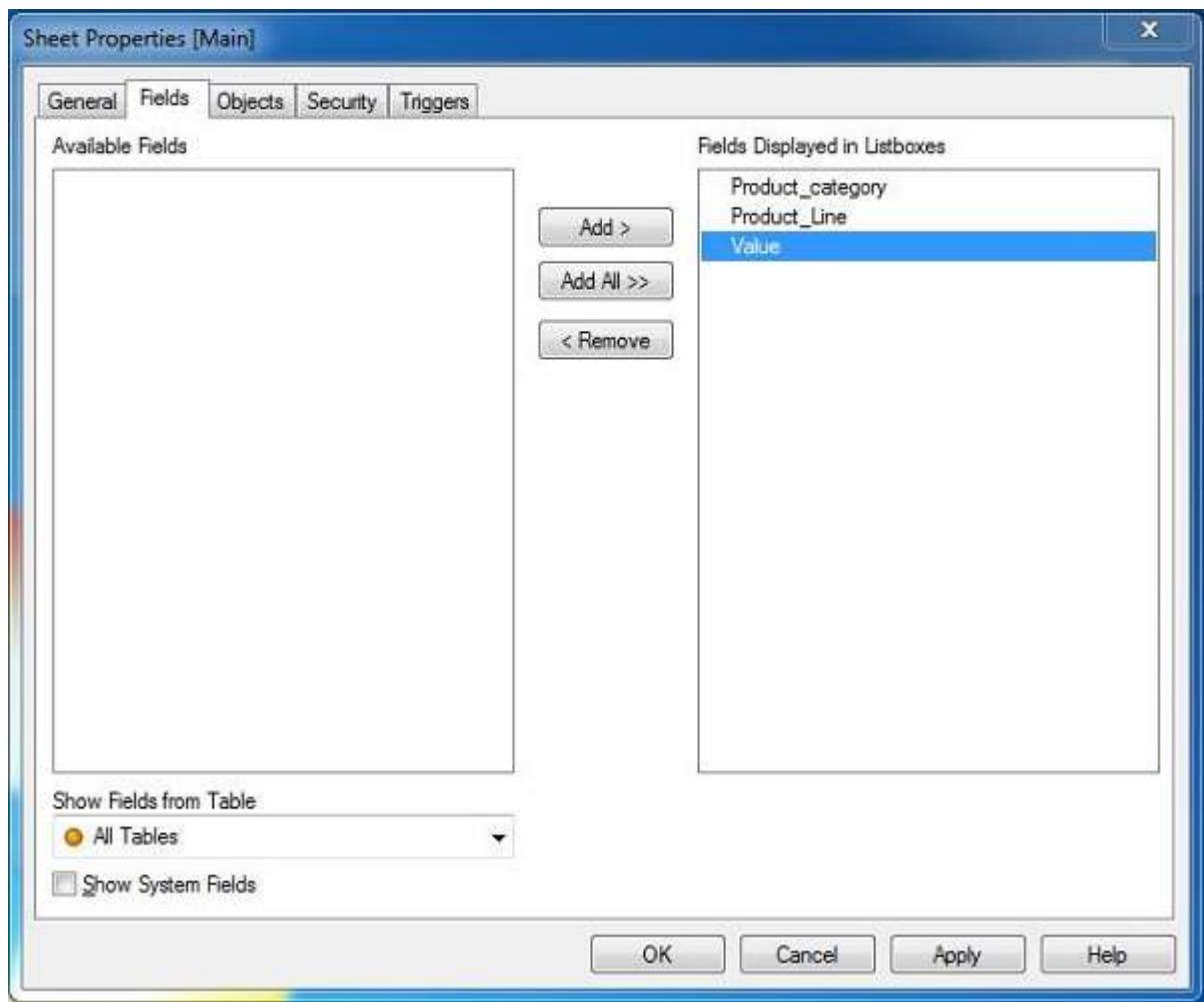
```

## Choose Matrices

We choose the fields from the above input data as matrices to be displayed in the dashboard. For this, we follow the steps in the menu **Layout -> Select Fields**.



In the next screen, choose the available fields to be displayed in the dashboard. Click "OK".



The following screen appears displaying all the fields.

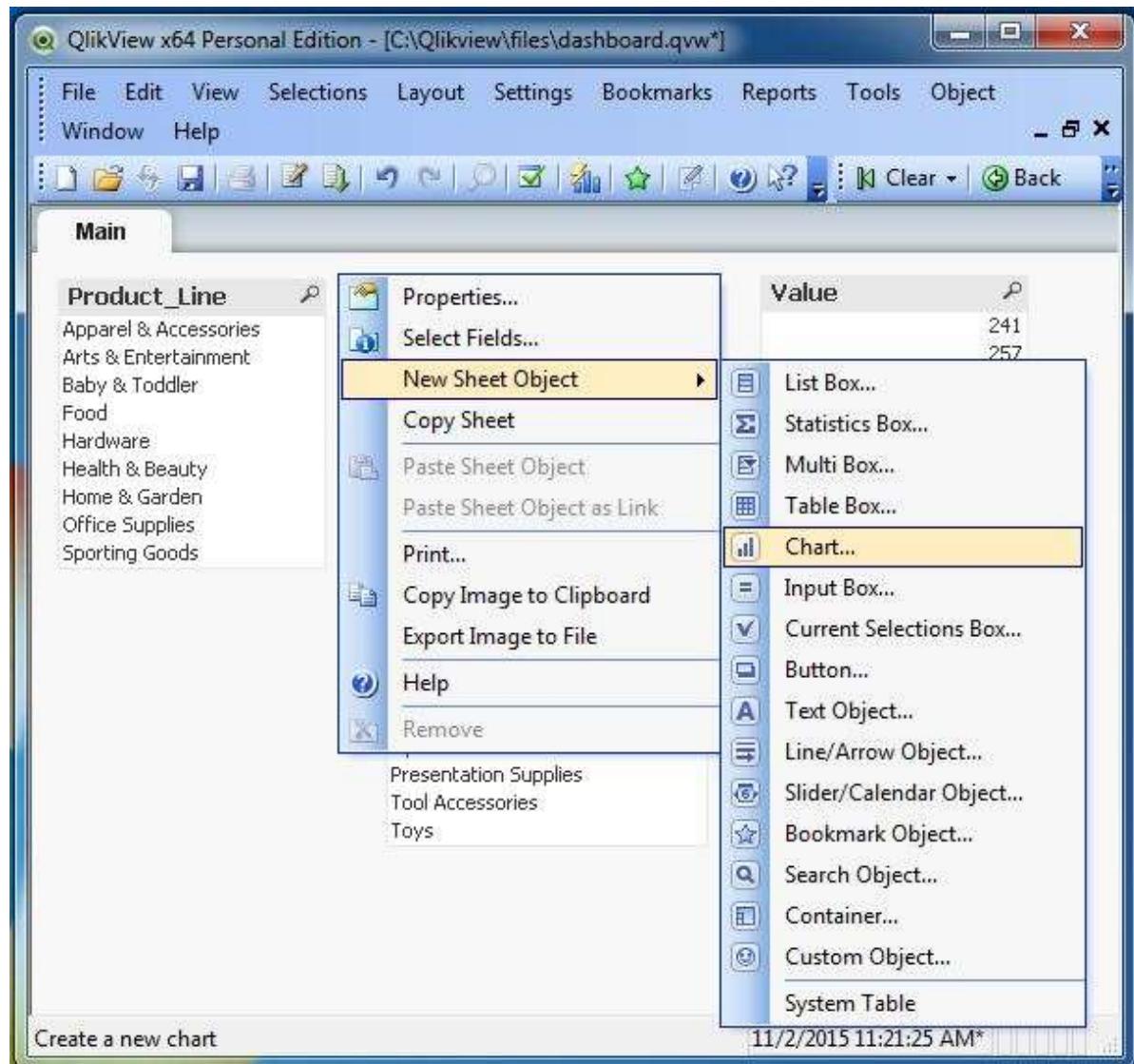
The screenshot shows the QlikView x64 Personal Edition interface with the title bar "QlikView x64 Personal Edition - [C:\Qlikview\files\dashboard.qvw\*]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. The toolbar below has various icons for file operations like Open, Save, Print, and search. The main area is titled "Main" and contains three data tables:

Product_Line	Product_category	Value
Apparel & Accessories	Athletics	241
Arts & Entertainment	Bathroom Accessories	257
Baby & Toddler	Beverages & Tobacco	456
Food	Blocks	548
Hardware	Clothing	577
Health & Beauty	Costumes & Accessories	812
Home & Garden	Diapering	1241
Office Supplies	Dispaly Borad	1245
Sporting Goods	Drinks	1247
	Hobbies & Creative Arts	2177
	Lawn & Garden	2365
	Musical Instruments	2514
	Outdoor Recreation	4487
	Paintings	5201
	Personal Care	5462
	Pipes	5642
	Presentation Supplies	6912
	Tool Accessories	8451
	Toys	

At the bottom, there is a status bar with "For Help, press F1", the date and time "11/2/2015 11:21:25 AM\*", and document statistics "D: 18/18 F: 19/19".

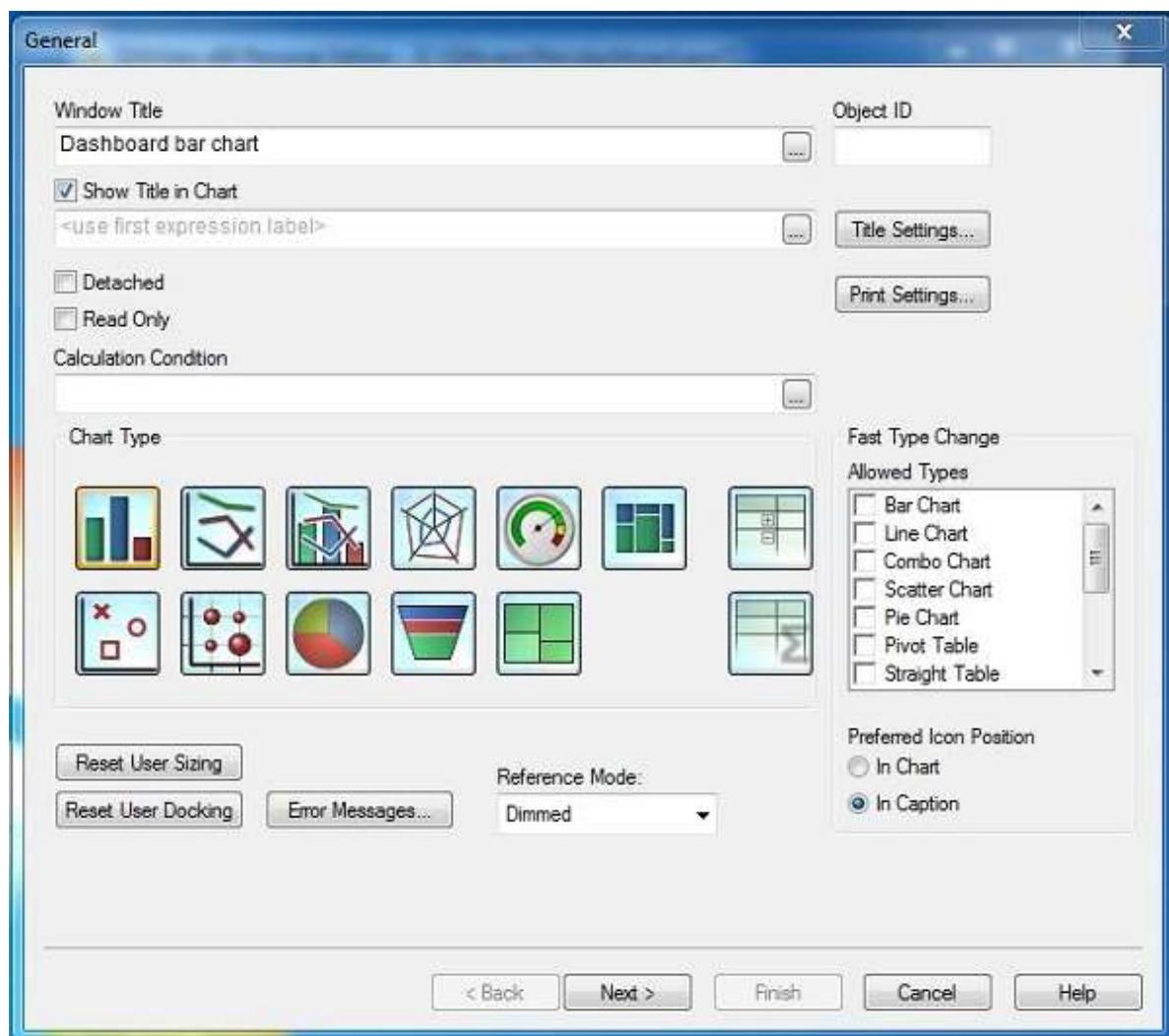
## Adding Chart to Dashboard

Now we add a chart to the dashboard by right-clicking anywhere in the sheet and choosing **New Sheet Object -> Chart**.



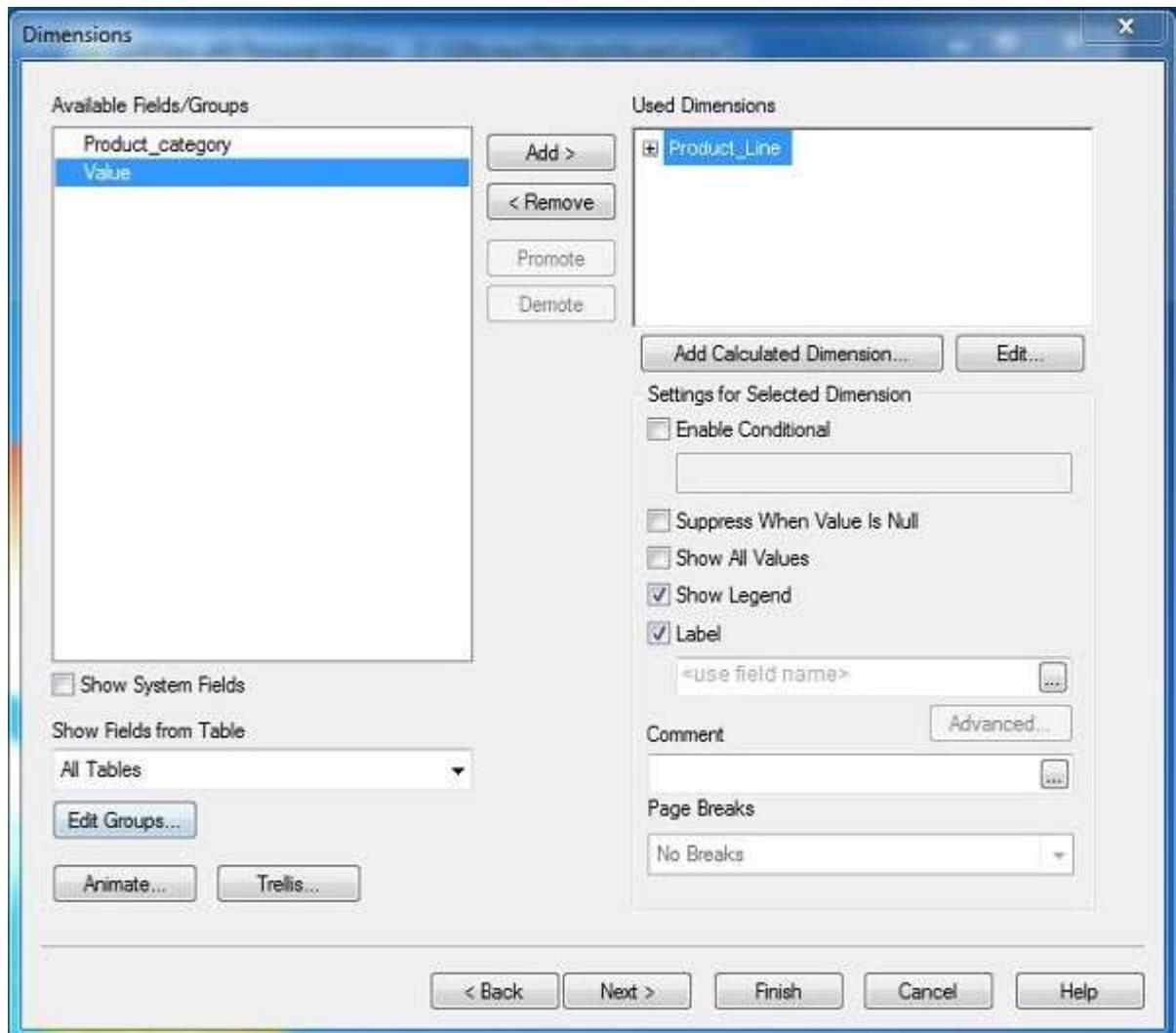
## Choose the Chart Type

Let us choose the chart type as a bar chart to display the sales values for various product Lines.



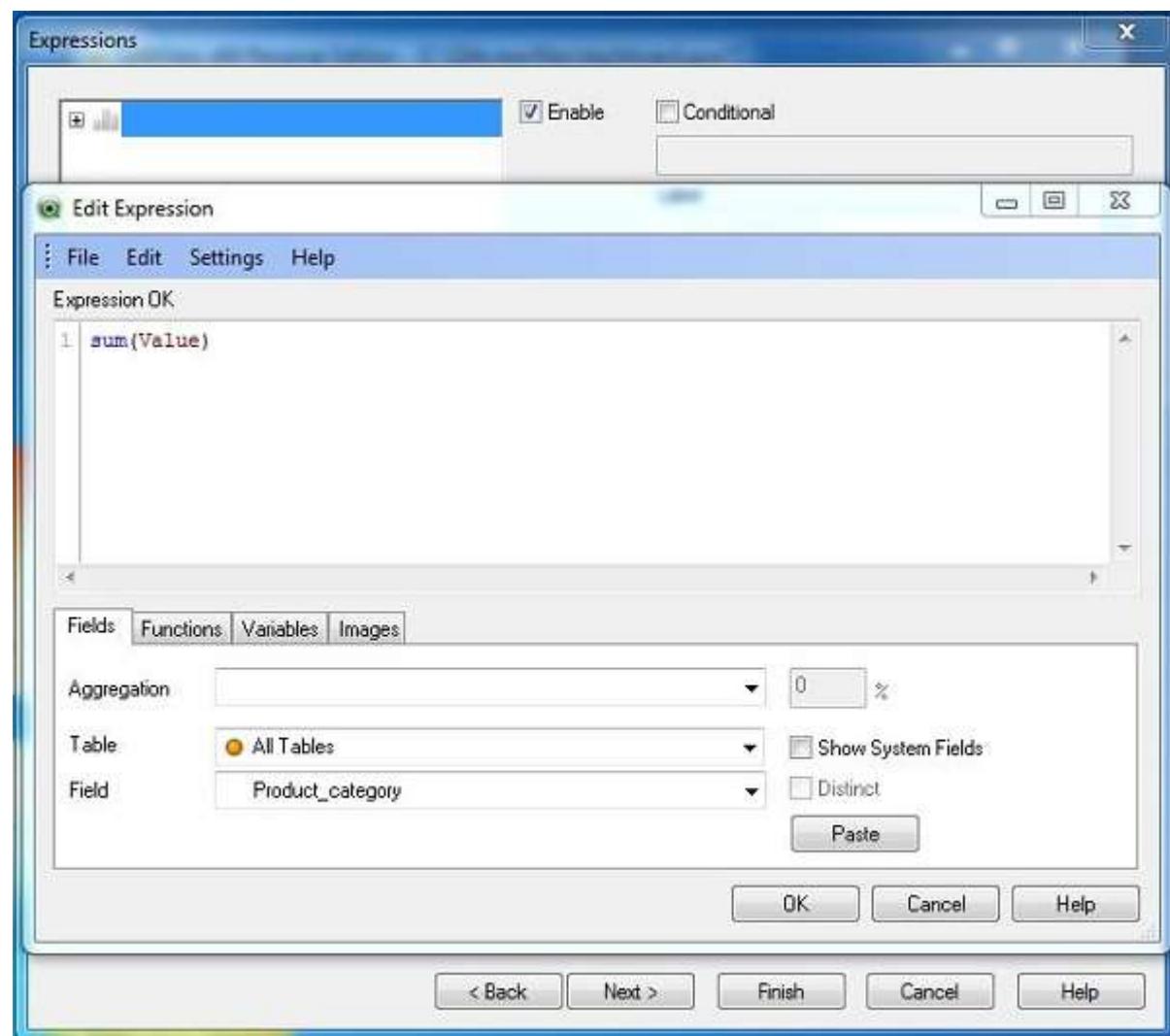
## Chart Dimension

Let us select the Product Line as the Chart Dimension.



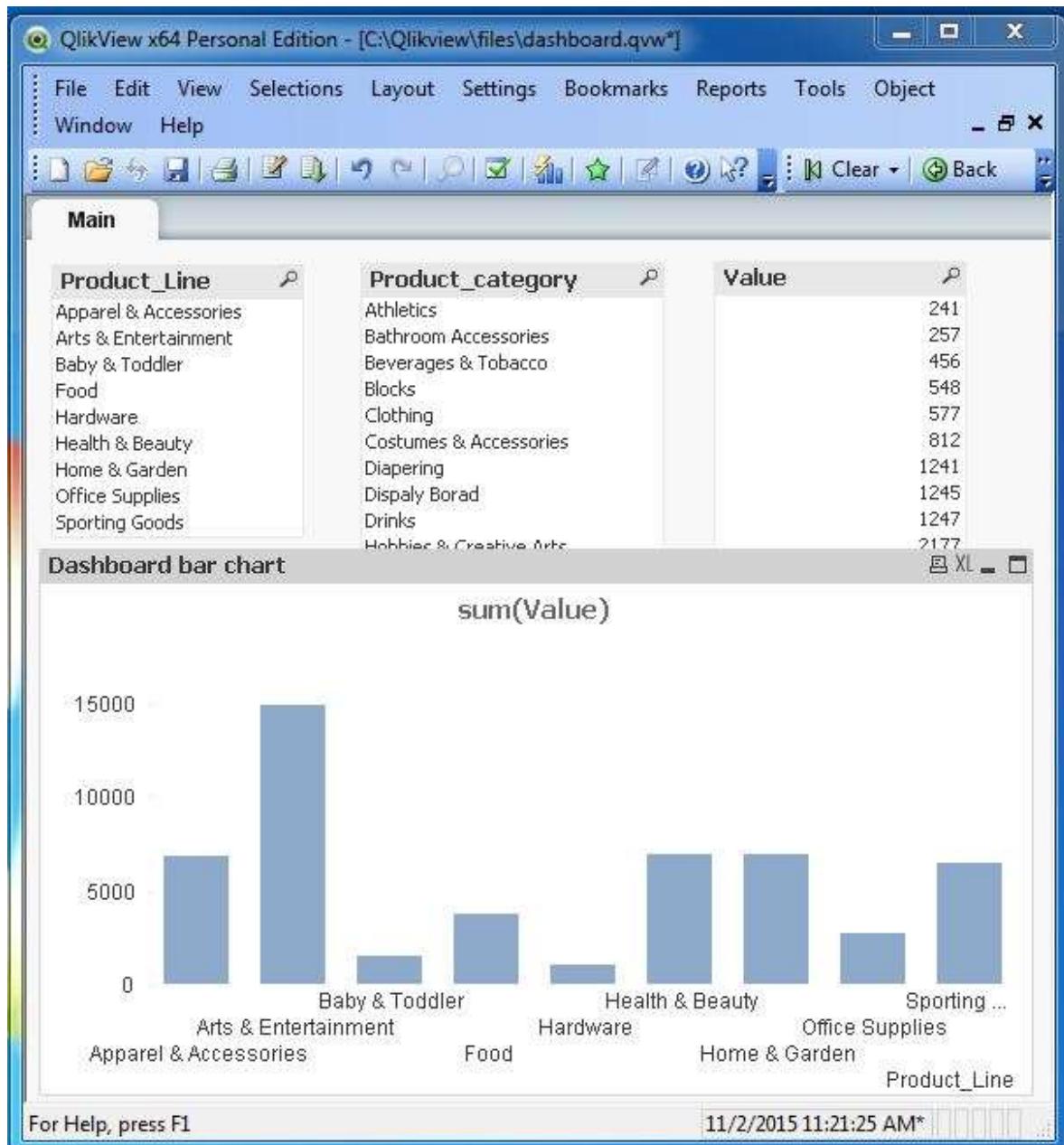
## Chart Expression

The expression to display the sales value for the Product Line dimension is written in the expression editor.



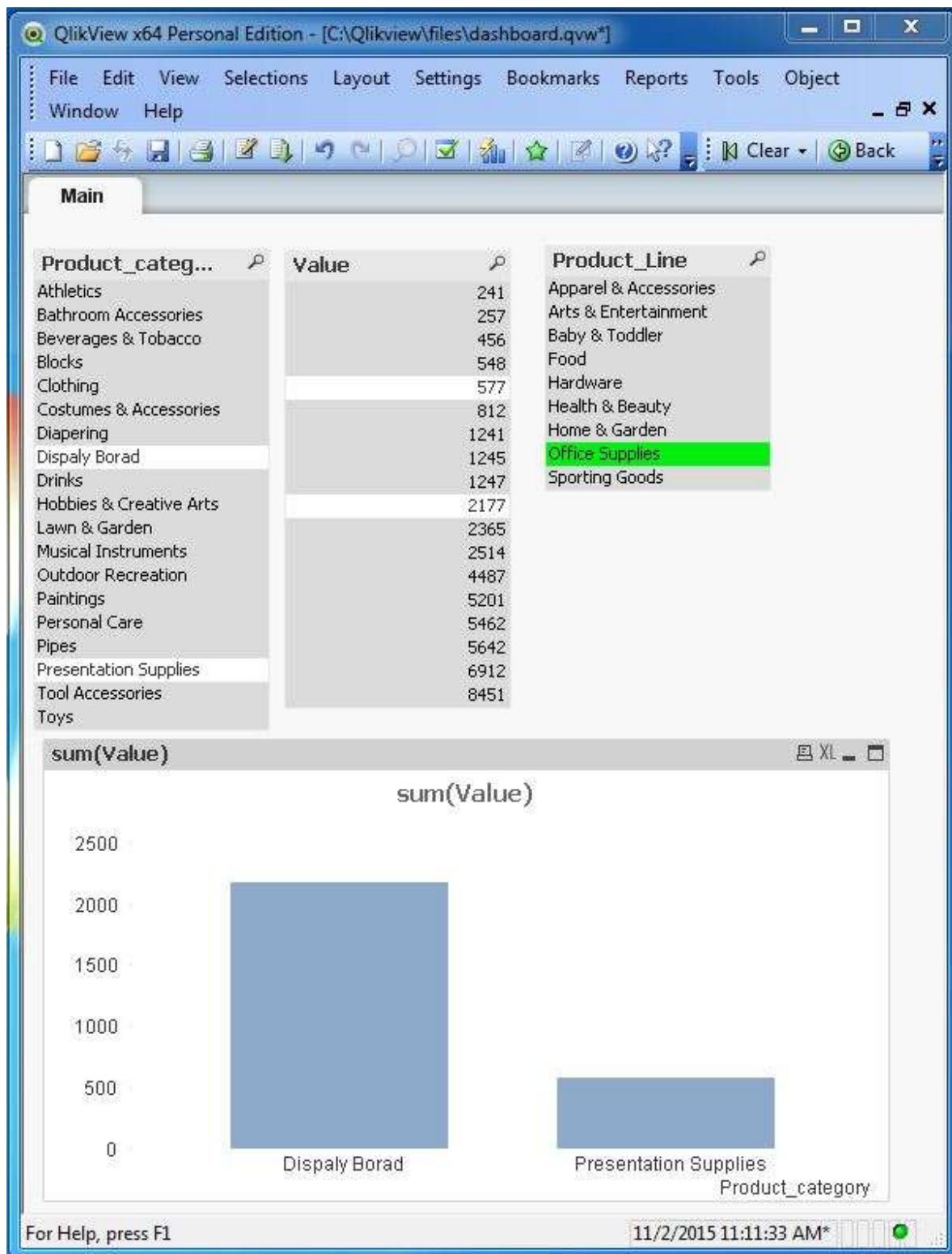
## The Dashboard

Given below is the dashboard displayed after finishing the above steps.



## Using the Dashboard

The values in the above Dashboard can be selected for filtering specific products and the chart changes accordingly. In addition, the associated values are highlighted.



# QlikView Data Transformation

## 31. QlikView – Data Transformation

Data Transformation is the process of modifying the existing data to a new data format. It can also involve filtering out or adding some specific values to the existing data set. QlikView can carry out data transformation after reading it to its memory and using many in-built functions.

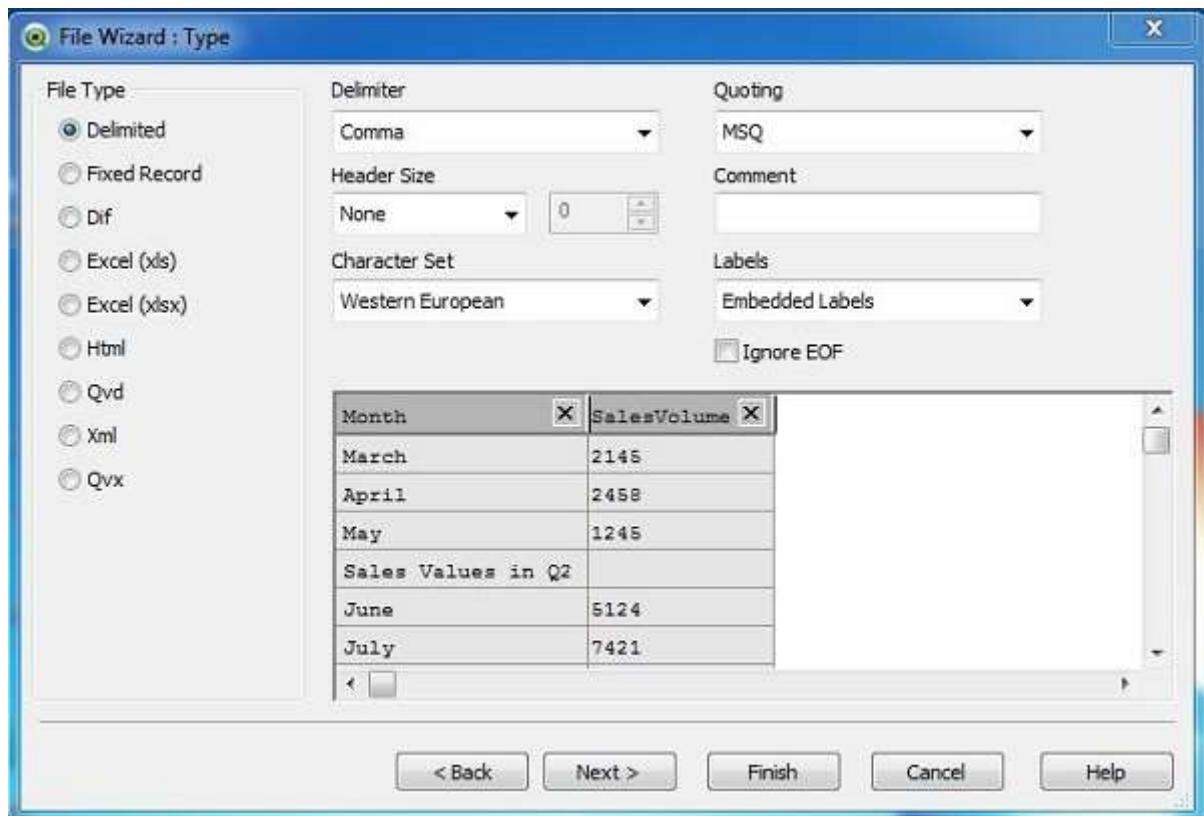
### Input Data

Let us consider the following input data, which represents the sales figures of each month. This is stored as a csv file with name quarterly\_sales.csv

```
Month,SalesVolume
March,2145
April,2458
May,1245
Sales Values in Q2
June,5124
July,7421
August,2584
Sales Values in Q3
September,5314
October,7846
November,6532
December,4625
January,8547
February,3265
```

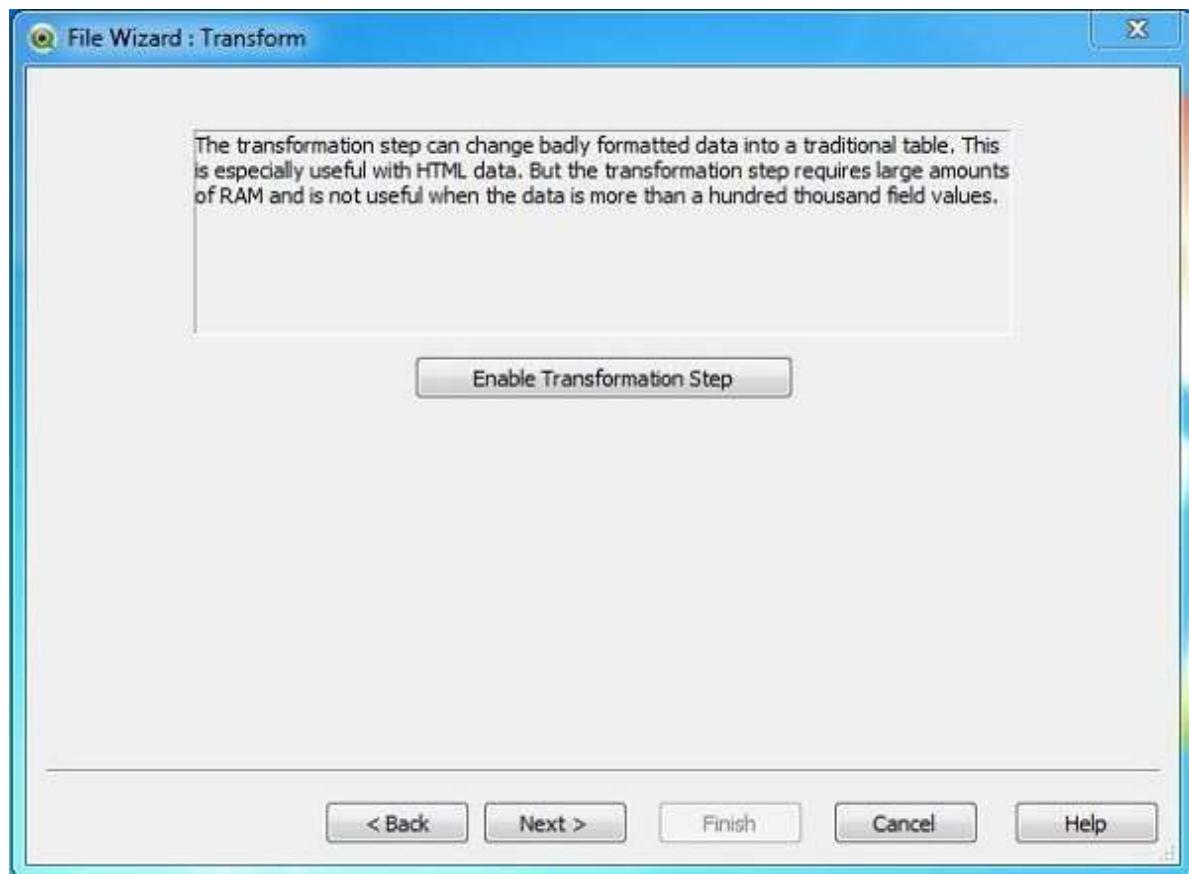
## Loading the Data

The above data is loaded to QlikView memory by using the script editor. Open the Script editor from the File menu or press **Control+E**. Choose the "Table Files" option form the "Data from Files" tab and browse for the file quarterlt\_sales.csv. Click next.



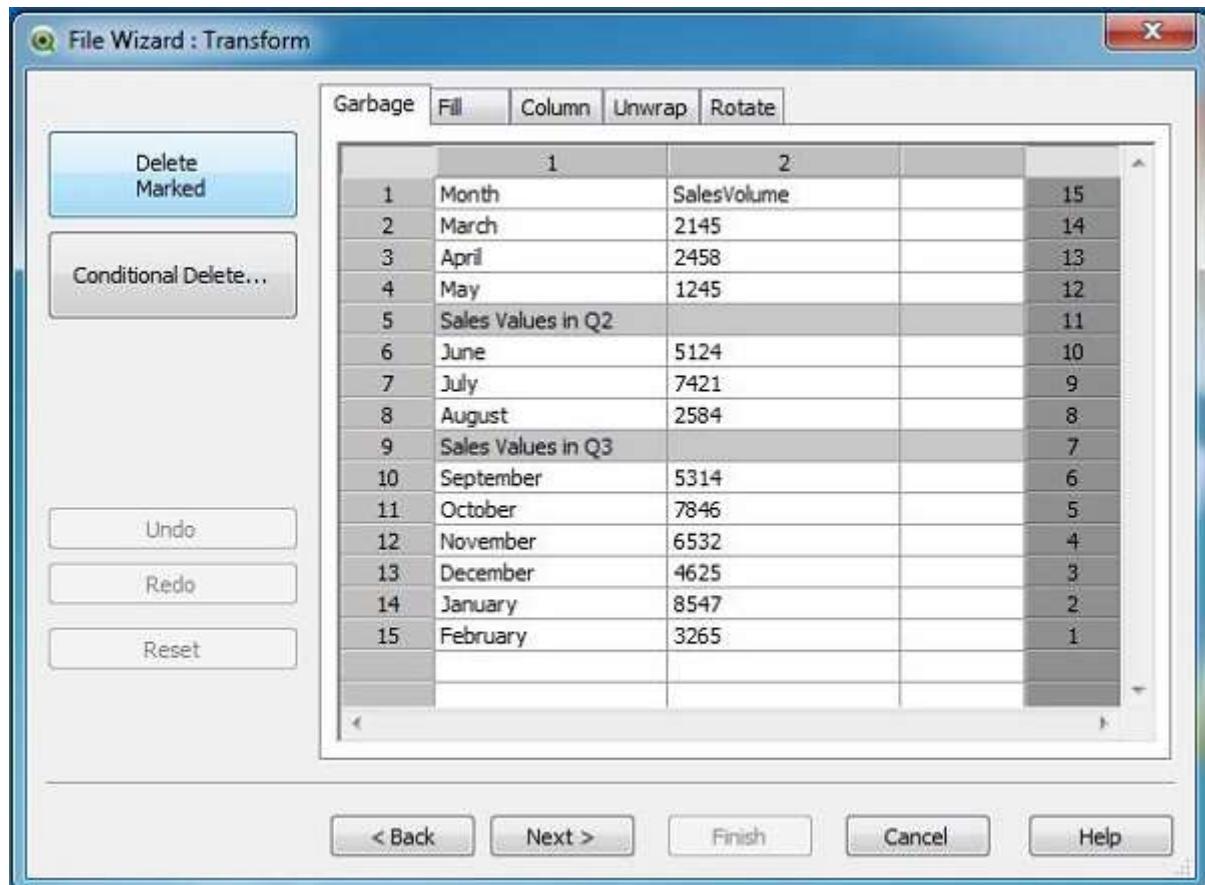
## The Transformation Wizard

The next screen prompts us to choose some data transformation. Click on the button **Enable Transformation Step**.



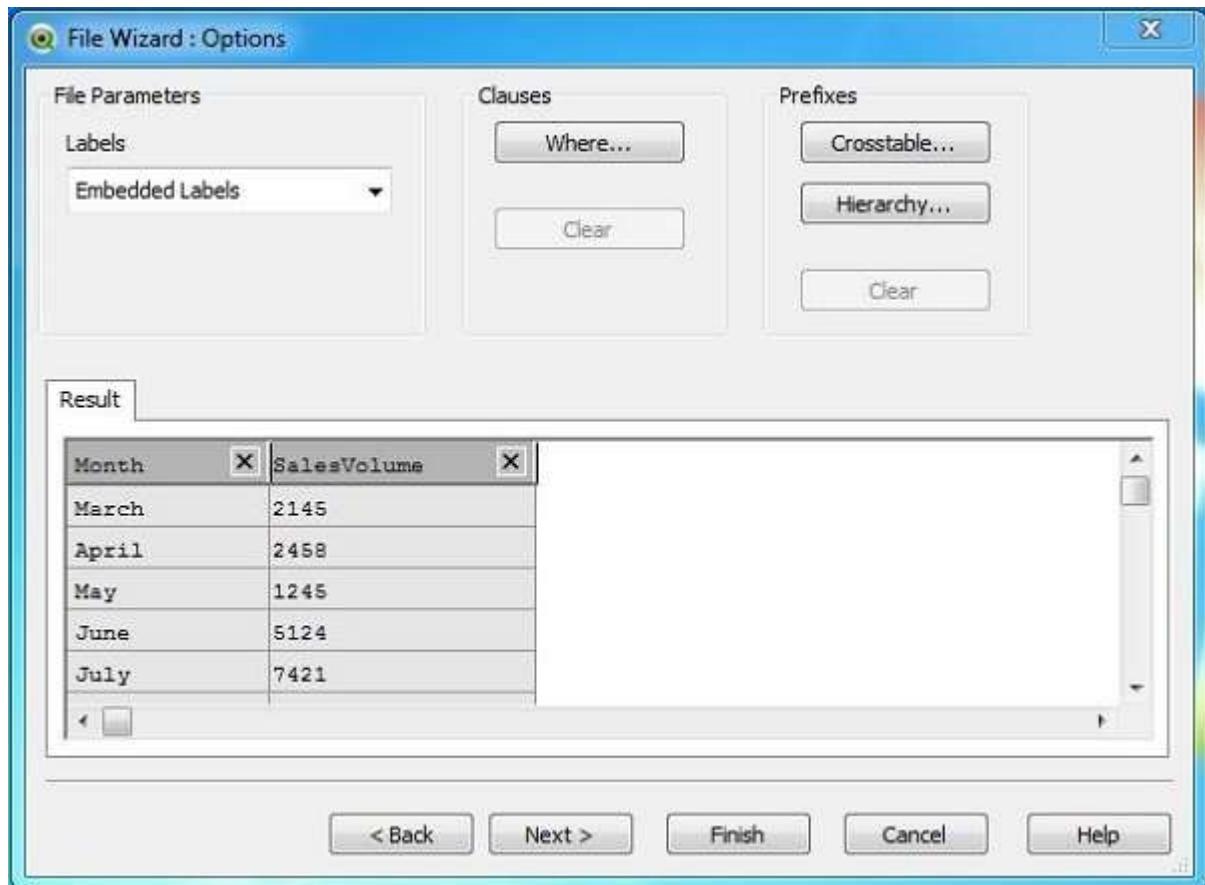
## Transformation Type

In this step, we will select the transformation to eliminate the rows, which describe the quarter. We select **Garbage -> delete marked** and select the two rows, which are not required. Click Next.



## File Wizard Options

After selecting the type of Transformation and the rows to be removed, the next screen prompts us for any further transformation like selecting a where clause or adding any Prefixes. We will ignore this step and click Finish.



## Load Script for Transformed Data

The Load script for the above data after all the transformation steps are complete is given below.

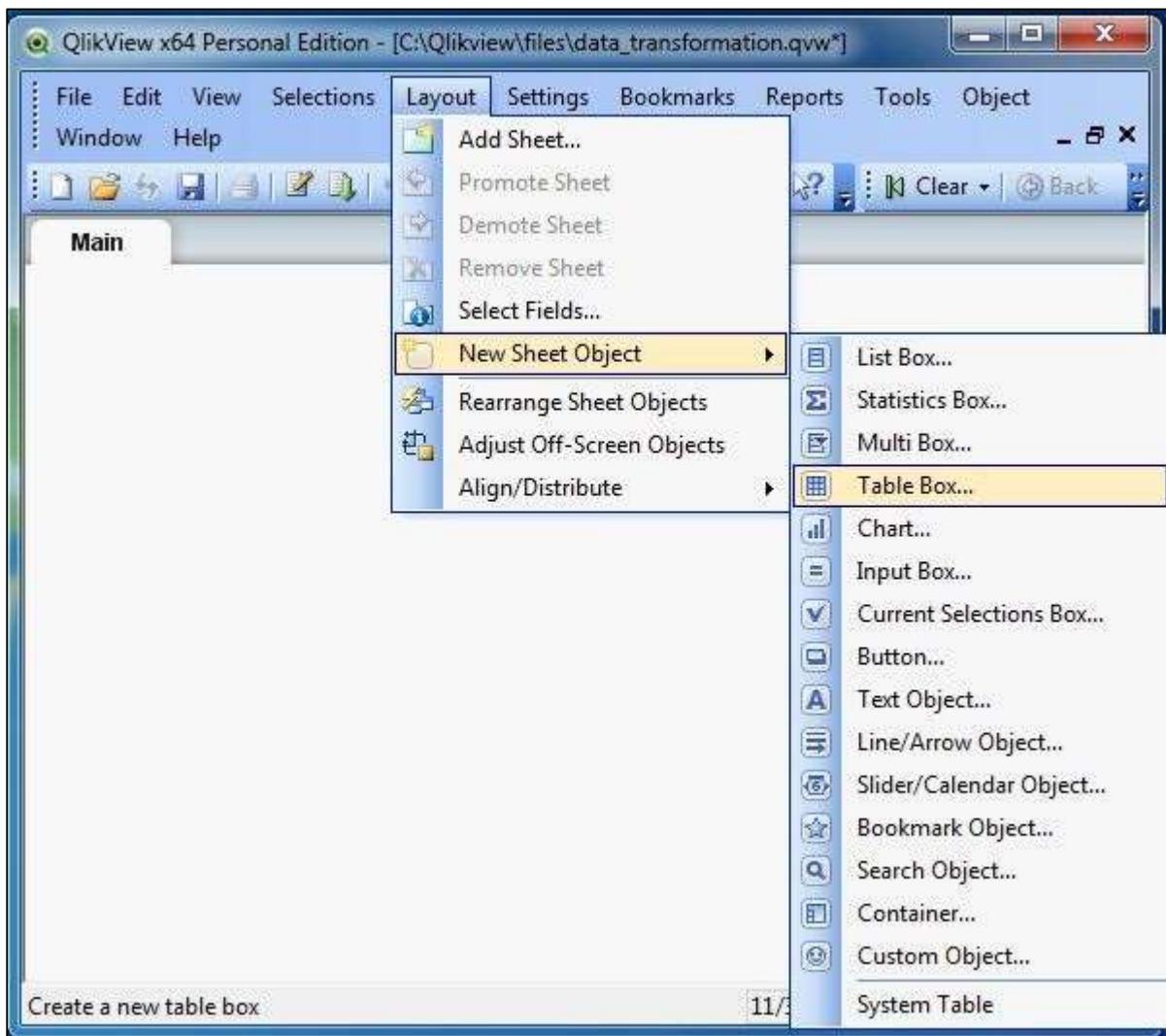
The screenshot shows the 'Edit Script [QV2]' dialog box. The main area contains the following QlikView load script:

```
1
2
3 LOAD Month,
4     SalesVolume
5 FROM
6 [C:\Qlikview\data\quarterly_sales.csv] .
7 (txt, codepage is 1252, embedded labels, delimiter is ',',,
8 msq, filters(
9 Remove(Row, Pos(Top, 9)),
10 Remove(Row, Pos(Top, 5))
11 ));
12
13
14
15
16
```

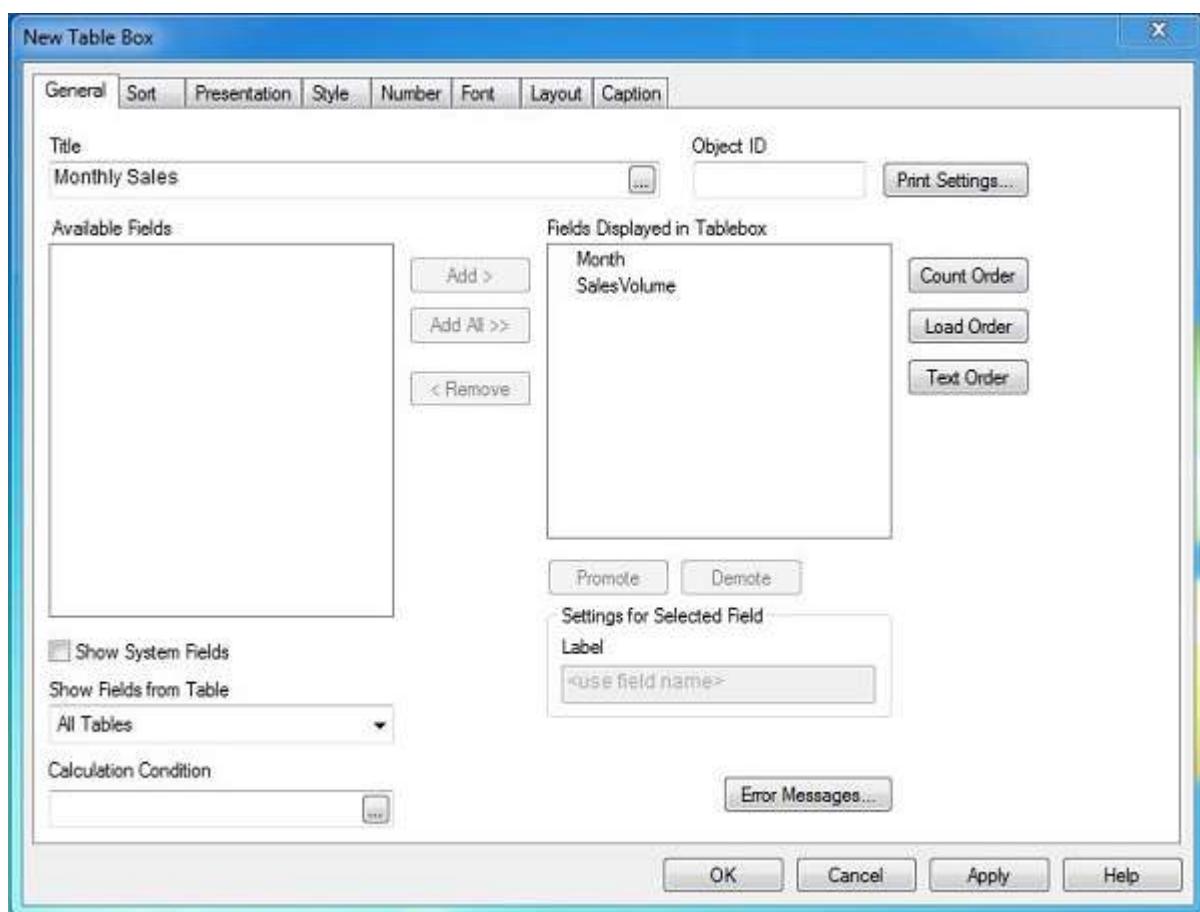
Below the script, there are tabs for Data, Functions, Variables, and Settings. The Data tab is selected, showing options for connecting to a Database (ODBC) and selecting Data from Files (Table Files, QlikView File, Web Files, Field Data). At the bottom are OK, Cancel, and Help buttons.

## Displaying Transformed Data

The transformed data can be displayed by using a **Table Box** sheet object. The steps to create it are given below.



Next, we choose the fields for the Table Box.



The Table Box now displays the data in the sheet.

The screenshot shows the QlikView x64 Personal Edition interface. The title bar reads "QlikView x64 Personal Edition - [C:\Qlikview\files\data\_transformation.qvw\*]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. The toolbar below the menu bar contains various icons for file operations like Open, Save, Print, and search. The main workspace is titled "Main" and contains a table titled "Monthly Sales". The table has two columns: "Month" and "SalesVolume". The data is as follows:

Month	SalesVolume
April	2458
August	2584
December	4625
February	3265
January	8547
July	7421
June	5124
March	2145
May	1245
November	6532
October	7846
September	5314

At the bottom of the screen, there is a status bar with the text "For Help, press F1", the date and time "11/3/2015 8:05:56 AM\*", and a window size indicator "12 X 2".

## 32. QlikView – Fill Function

The **Fill** function in QlikView is used to fill values from existing fields into a new field.

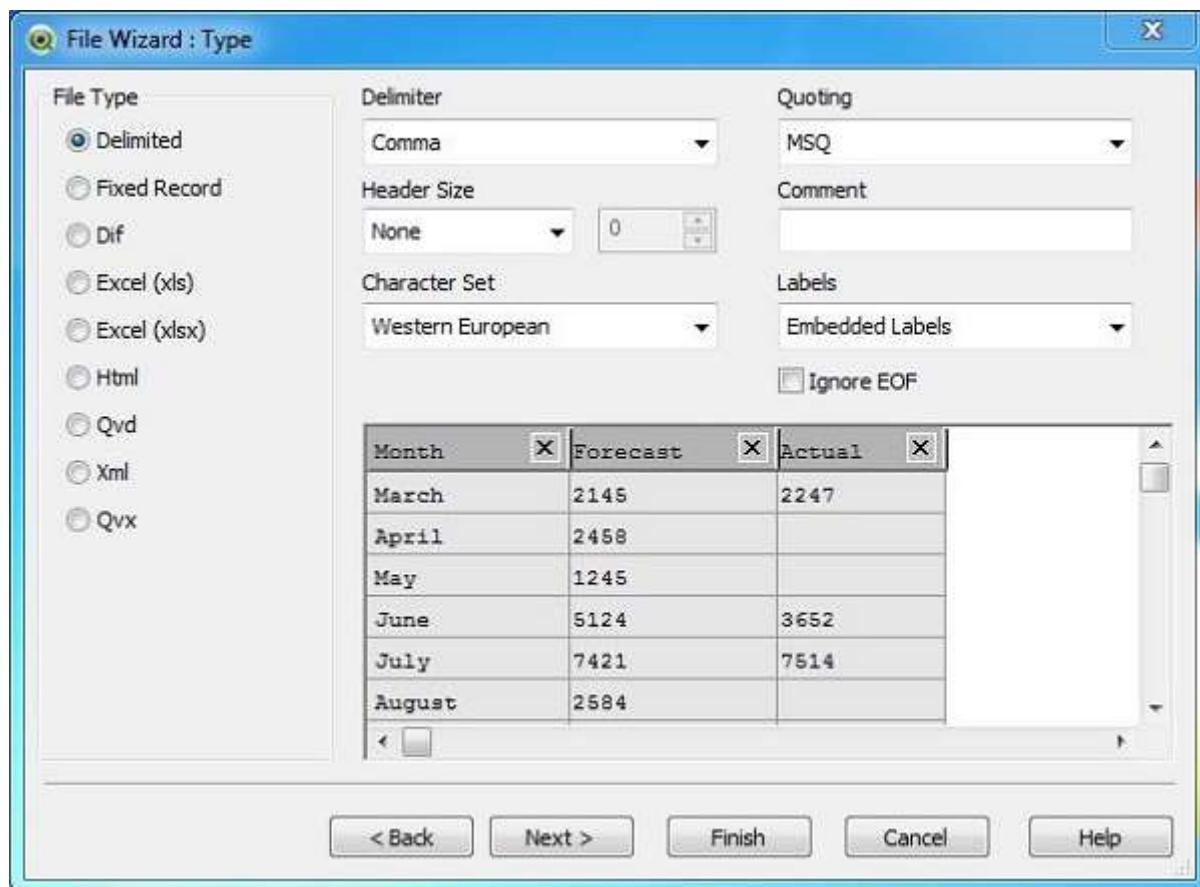
### Input Data

Let us consider the following input data, which represents the actual and forecasted sales figures.

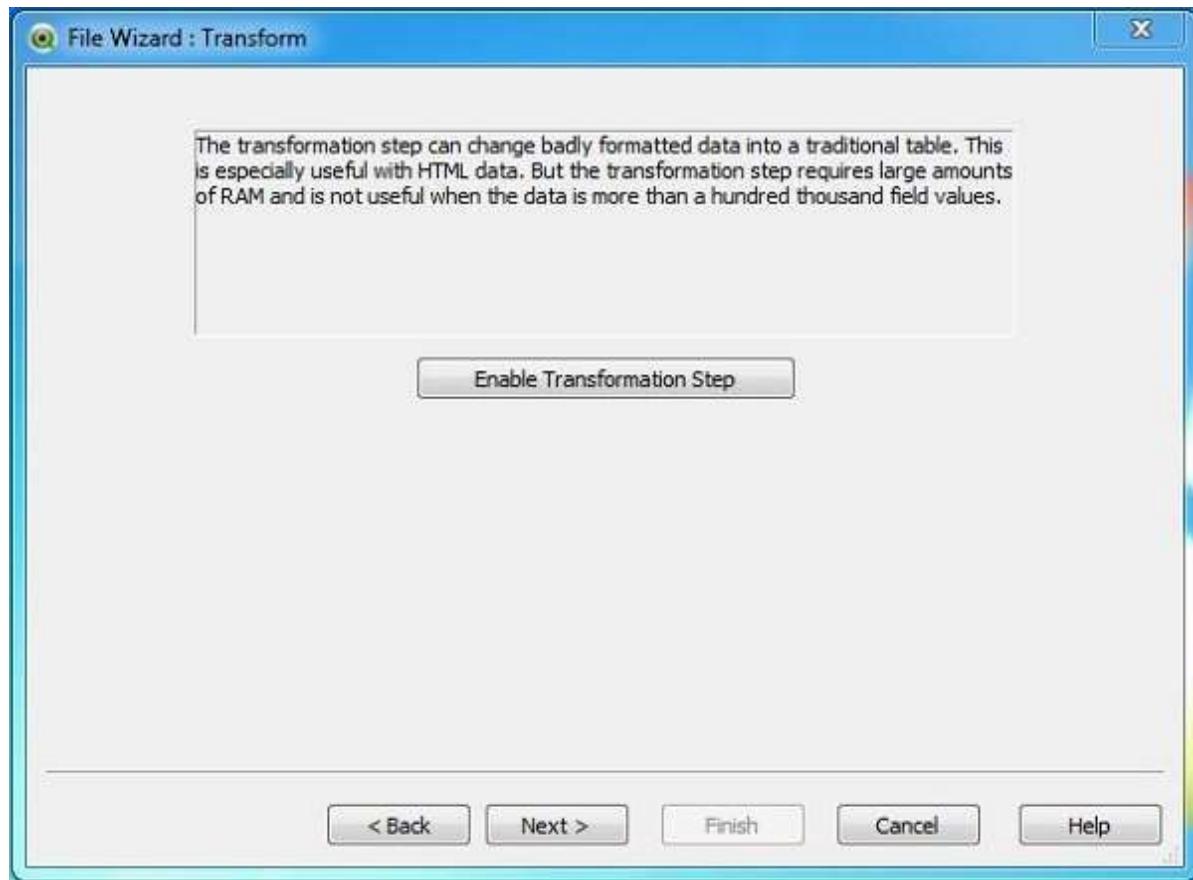
```
Month,Forecast,Actual
March,2145,2247
April,2458,
May,1245,
June,5124,3652
July,7421,7514
August,2584,
September,5314,4251
October,7846,6354
November,6532,7451
December,4625,1424
January,8547,7852
February,3265,
```

## Load Script

The above data is loaded to the QlikView memory by using the script editor. Open the Script editor from the File menu or press **Control+E**. Choose the "Table Files" option from the "Data from Files" tab and browse for the file containing the above data.



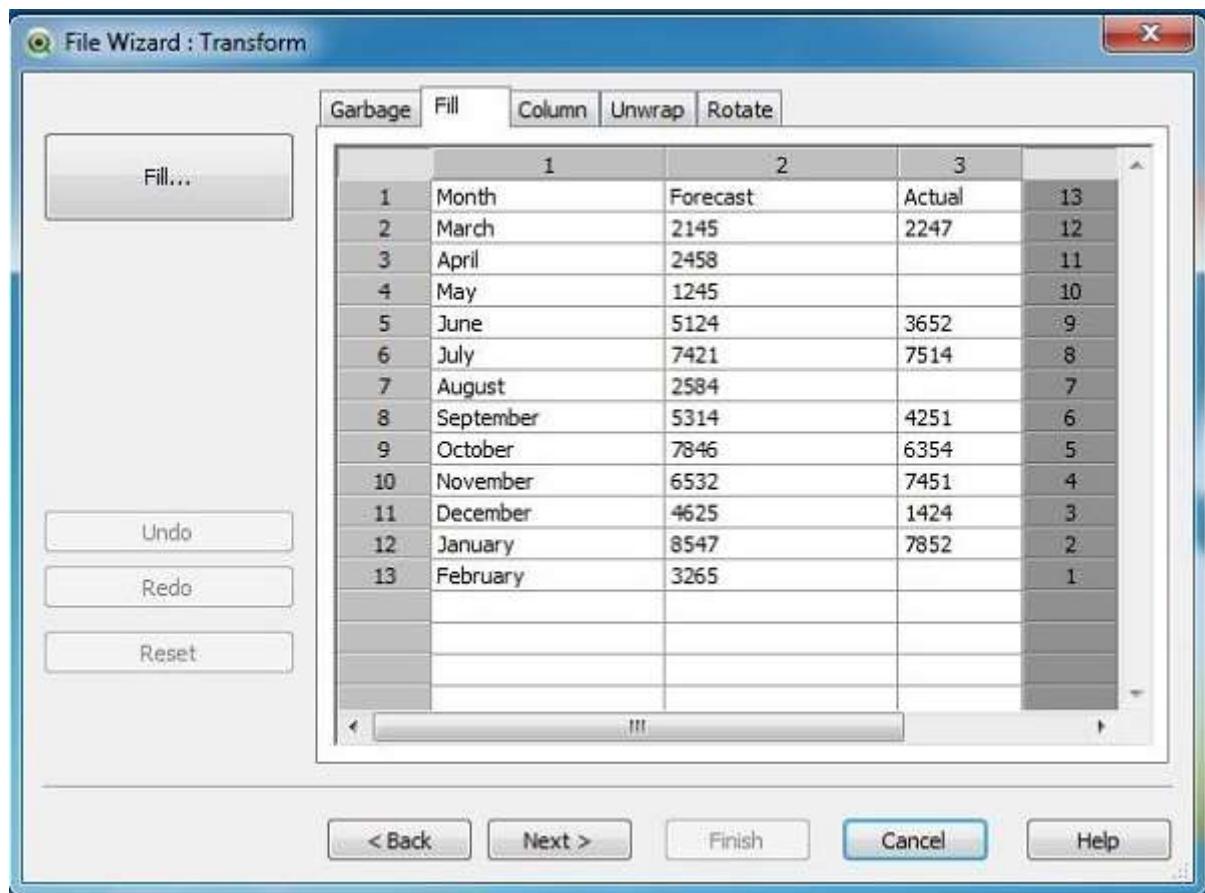
After clicking Next in the above step, we choose the **Enable Transformation Step** button to carry out the required data transformation.



## Selecting the Data Transformation

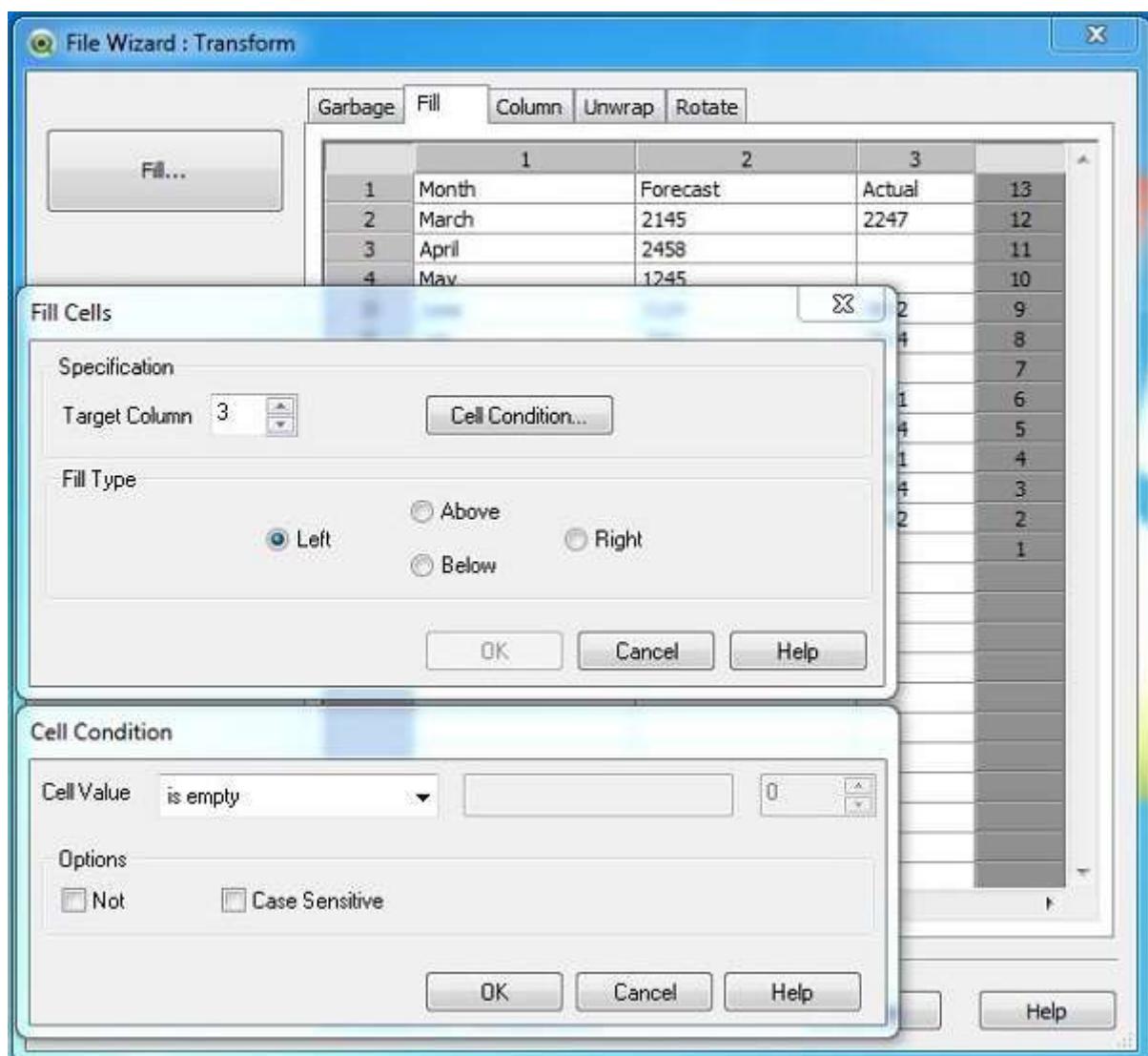
179

As we are going to use the **Fill** function, let us choose the Fill tab, which displays the empty values under the Actual Field.



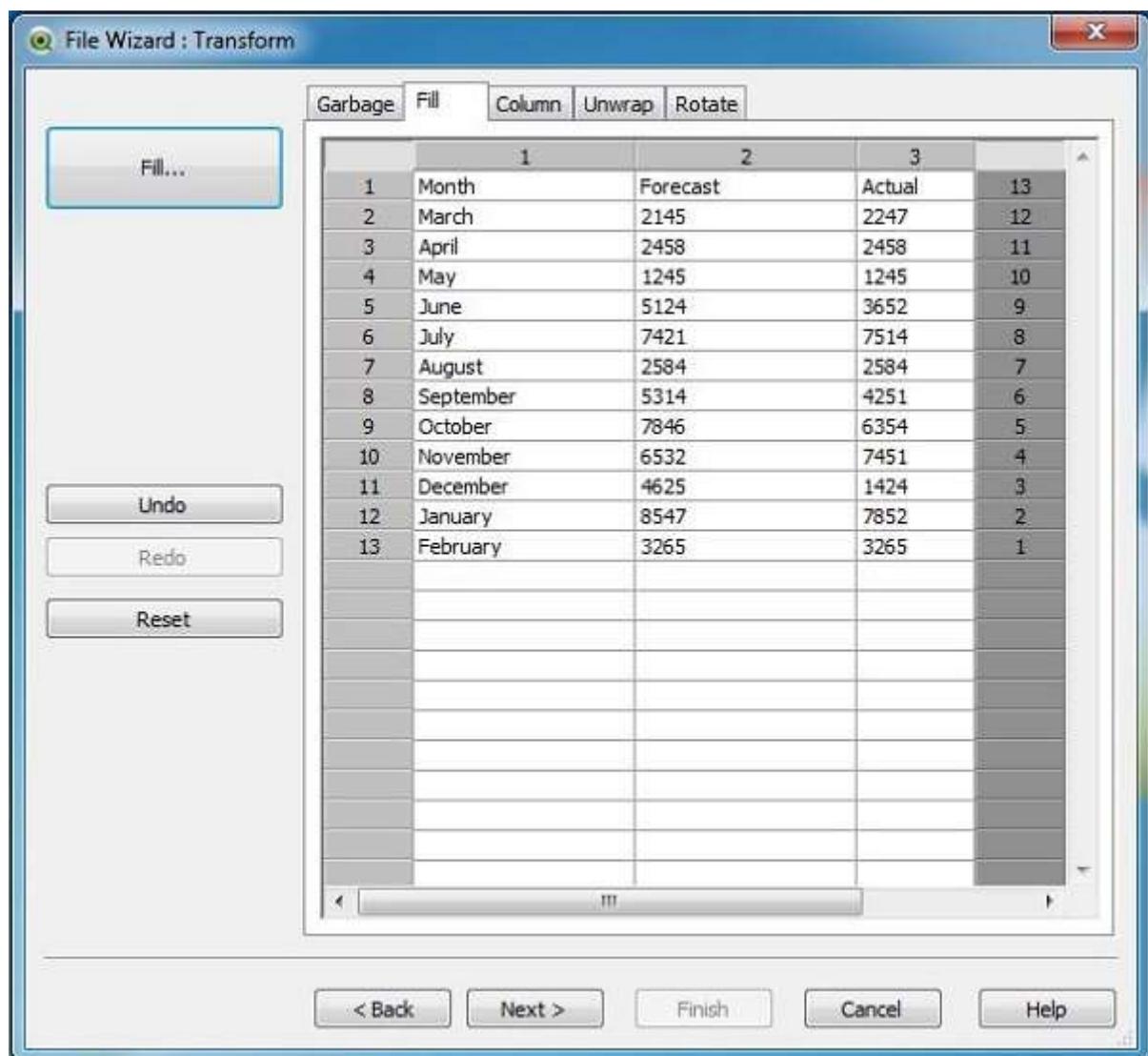
## Create the Fill Condition

On clicking the Fill button, the option to choose target column and the cell condition appears. We choose column three, as we want to fill the empty values of this column with values from same row in column two. Also, choose the Cell Value as empty so that only the empty cells will be overwritten with new values.



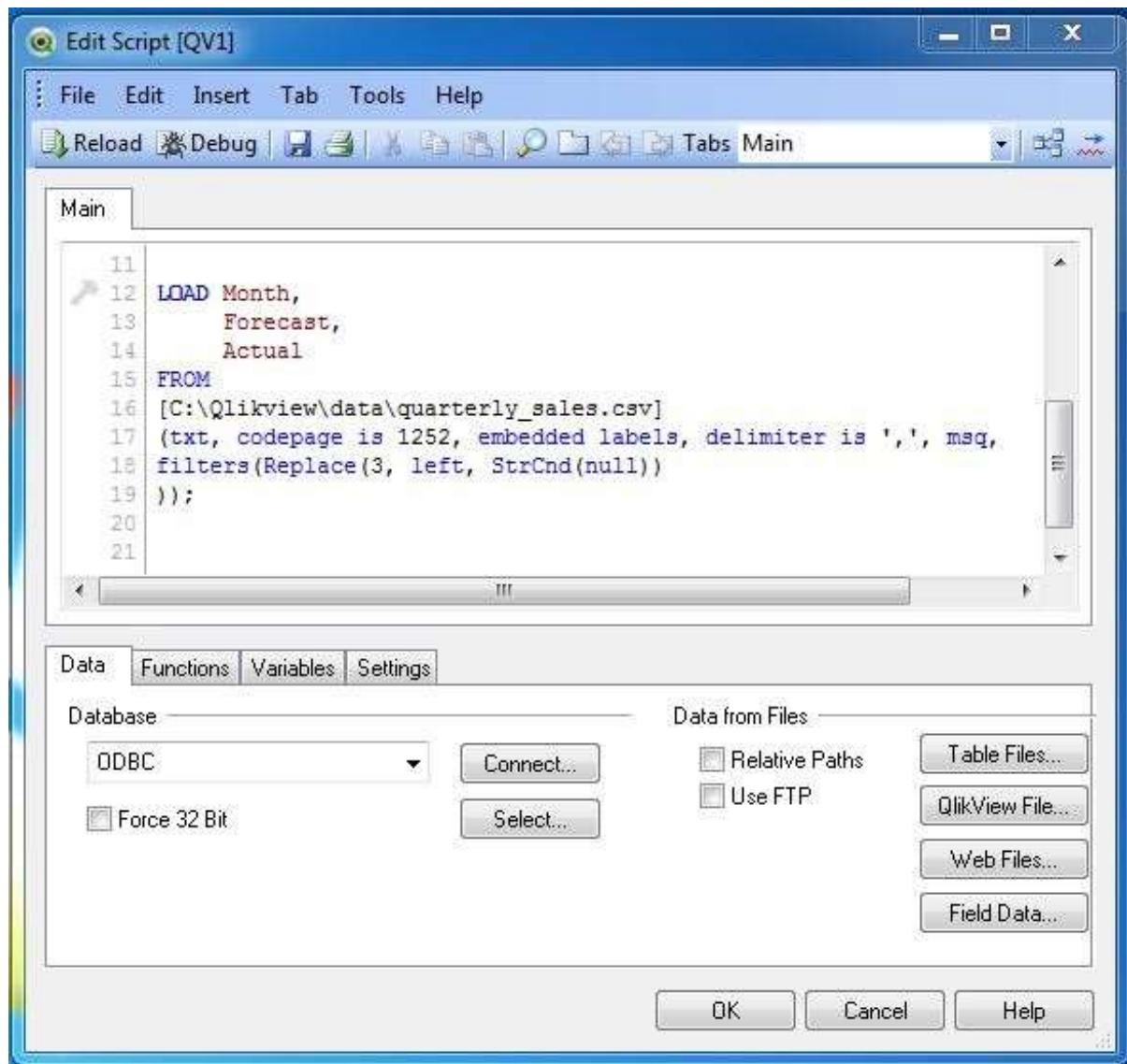
## Transformed Data

On completing the above steps, we get the transformed data as shown below.



## Load Script for Transformed Data

The load script for the transformed data can be seen using the script editor. The script shows the expression, which replaces the empty cell values.



## Display Transformed Data

The transformed data can be seen by creating a **Table Box** using the option in the menu **Layout -> New Sheet Object**.

QlikView x64 Personal Edition - [C:\Qlikview\files\fill\_func.qvw]

File Edit View Selections Layout Settings Bookmarks Reports Tools Object Window Help

Main

Sales Forecast

Month	Forecast	Actual
April	2458	2458
August	2584	2584
December	4625	1424
February	3265	3265
January	8547	7852
July	7421	7514
June	5124	3652
March	2145	2247
May	1245	1245
November	6532	7451
October	7846	6354
September	5314	4251

For Help, press F1 11/3/2015 7:37:27 PM\* 12 X 3

## 33. QlikView – Column Manipulation

Column Manipulation is a type of Data Transformation in which a new column is populated with values from an existing column, which meets certain criteria. The criteria can be an expression, which is created as part of the Data Transformation step.

### Input Data

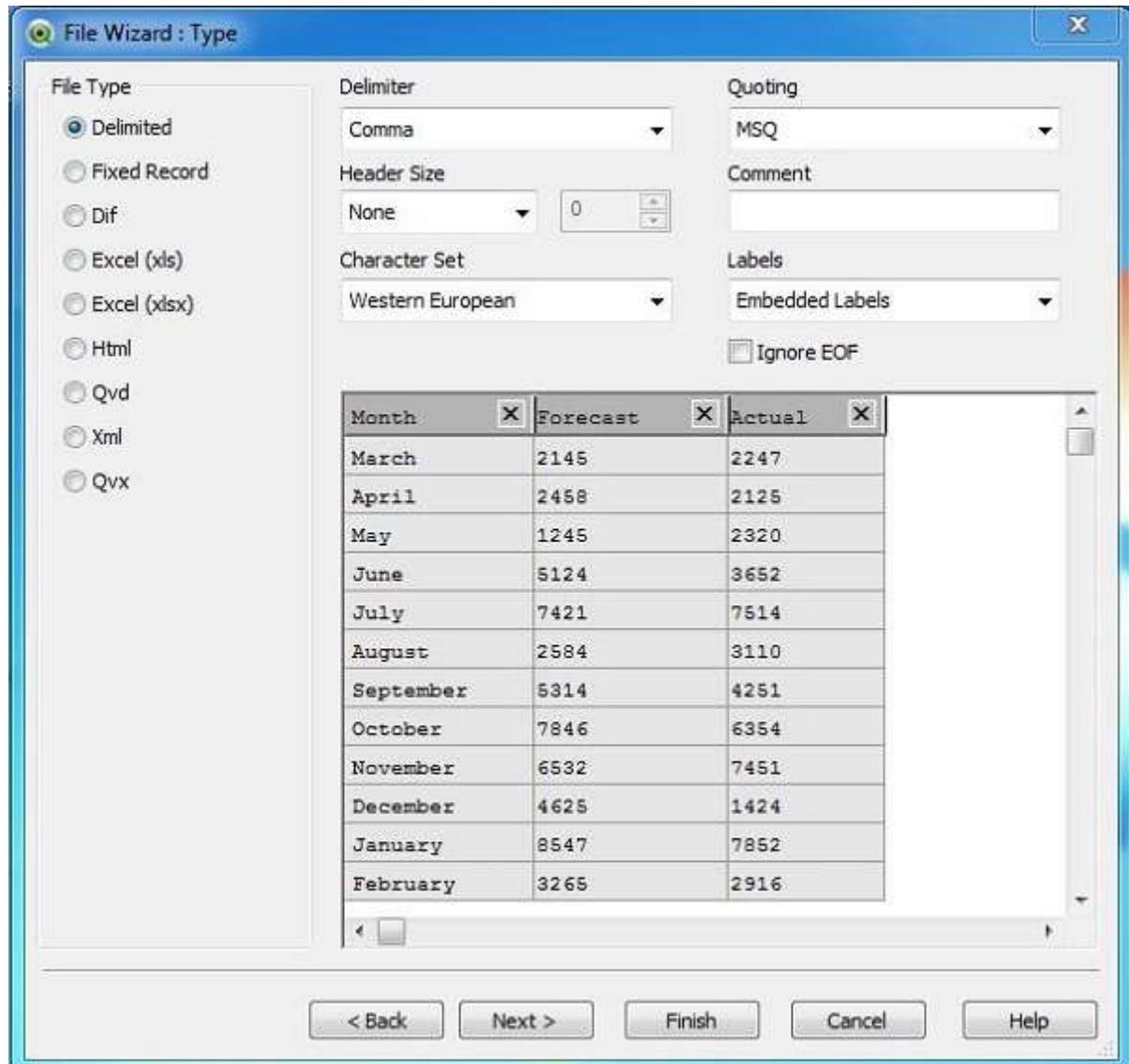
Let us consider the following input data, which represents the actual and forecasted sales figures.

```
Month,Forecast,Actual
March,2145,2247
April,2458,2125
May,1245,2320
June,5124,3652
July,7421,7514
August,2584,3110
September,5314,4251
October,7846,6354
November,6532,7451
December,4625,1424
January,8547,7852
February,3265,2916
```

### Load Script

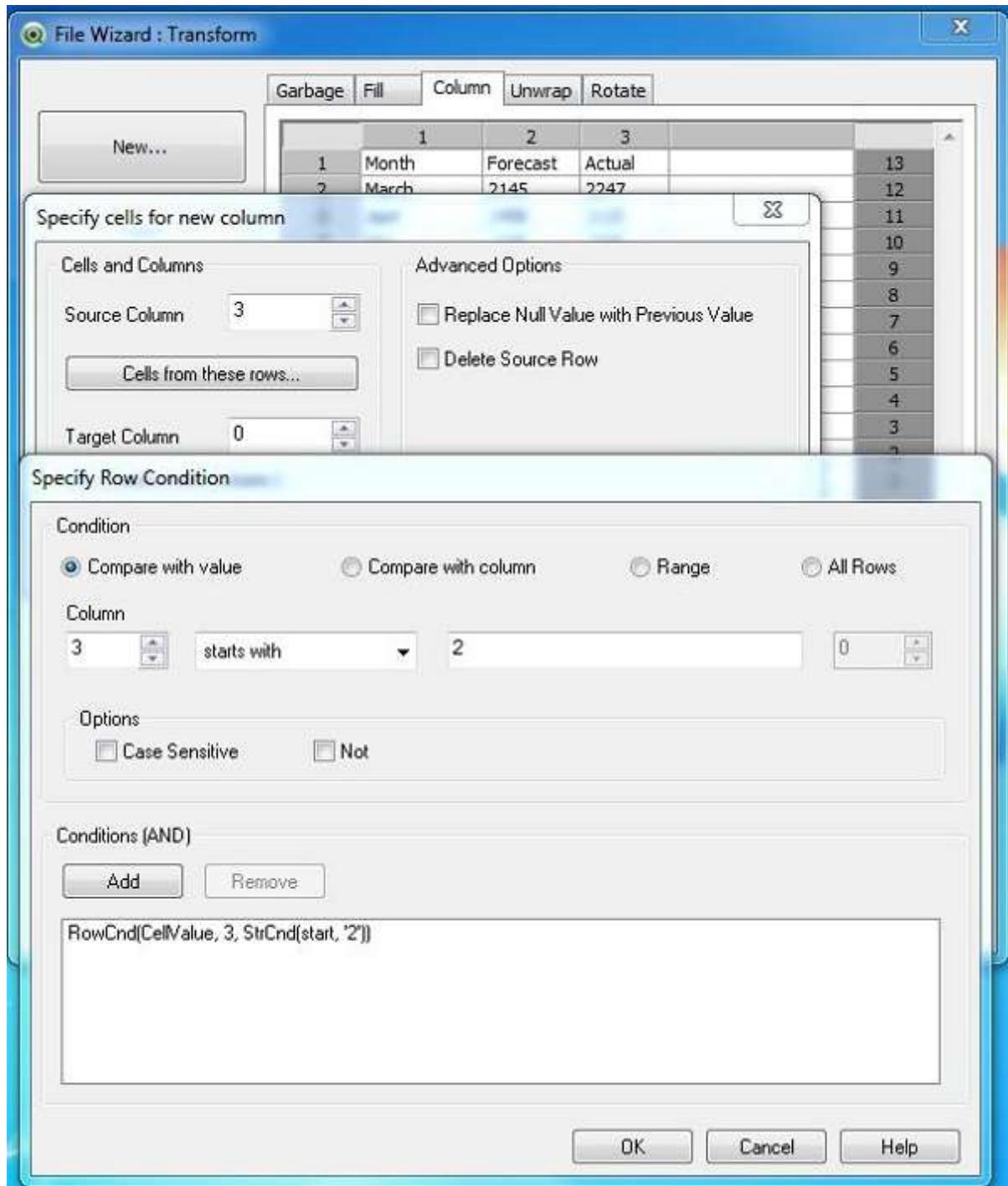
The above data is loaded to QlikView memory by using the script editor. Open the Script editor from the File menu or press **Control+E**. Choose the "Table Files" option from the

"Data from Files" tab and browse for the file containing the above data. After clicking Next, we choose the **Enable Transformation Step** button to carry out the required data transformation.



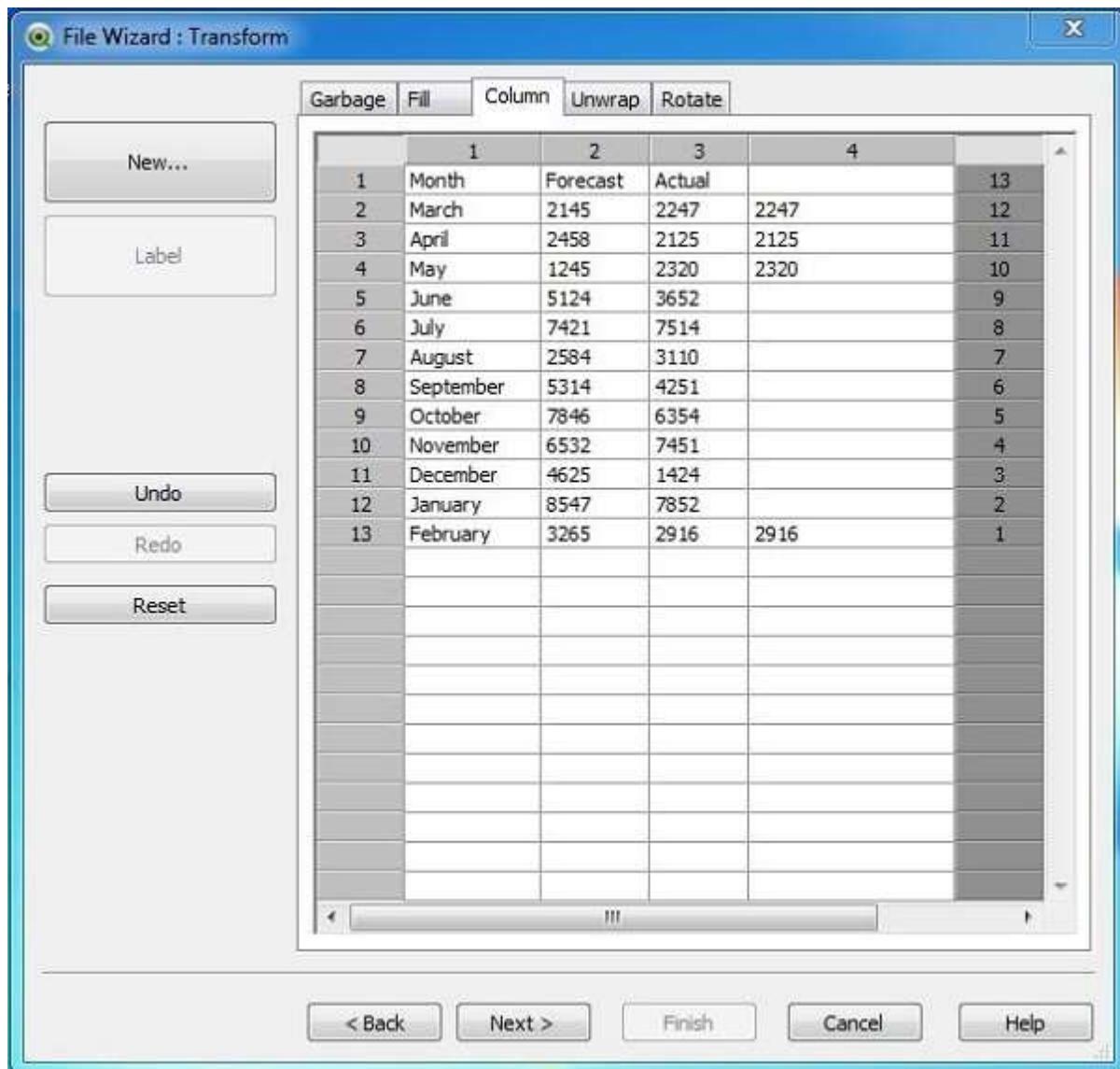
## Selecting the Data Transformation

Choose the Column tab and then choose the **New** button. It asks to specify the **New column** and the Row Condition. We specify column 3 as the source column and pick the values, which start with two as the Row Condition.



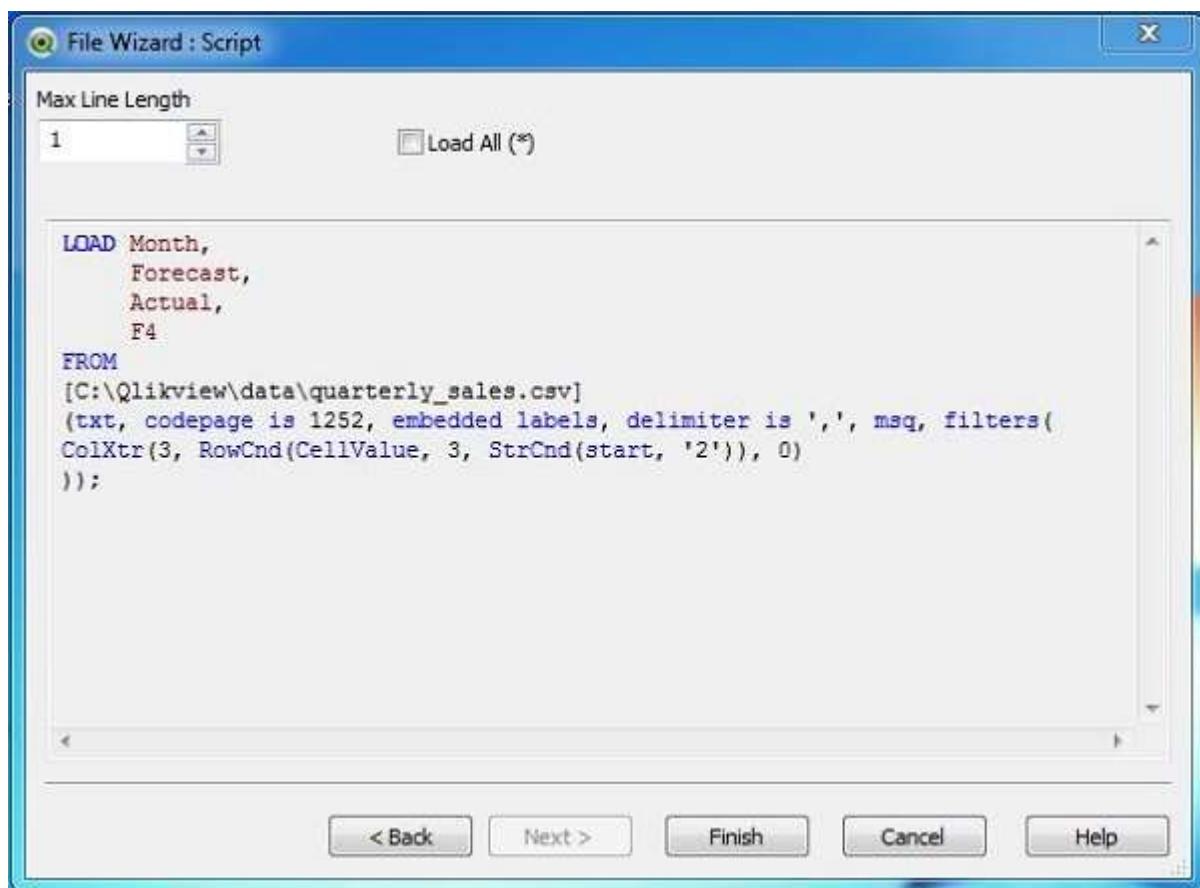
## Transformed Data

On completing the above steps, we get the transformed data as shown below.



## Load Script for Transformed Data

The load script for the Transformed data can be seen using the script editor. The script shows the expression, which creates the new column with required values.



## Display Transformed Data

The transformed data can be seen by creating a **Table Box** using the option in the menu **Layout -> New Sheet Object**.

QlikView x64 Personal Edition - [QV1]

File Edit View Selections Layout Settings Bookmarks Reports Tools Object Window Help

Clear Back

Main

Column Manipulation

Month	Forecast	Actual	F4
December	4625	1424	
April	2458	2125	2125
March	2145	2247	2247
May	1245	2320	2320
February	3265	2916	2916
August	2584	3110	
June	5124	3652	
September	5314	4251	
October	7846	6354	
November	6532	7451	
July	7421	7514	
January	8547	7852	

For Help, press F1

## 34. QlikView – Rotating Tables

The Rotating table in QlikView is similar to the column and row transpose feature in Microsoft Excel but with some additional options. We can transpose columns in multiple directions and they give different results. In this chapter, we will be seeing the normal transpose option of converting rows to columns.

### **Input Data**

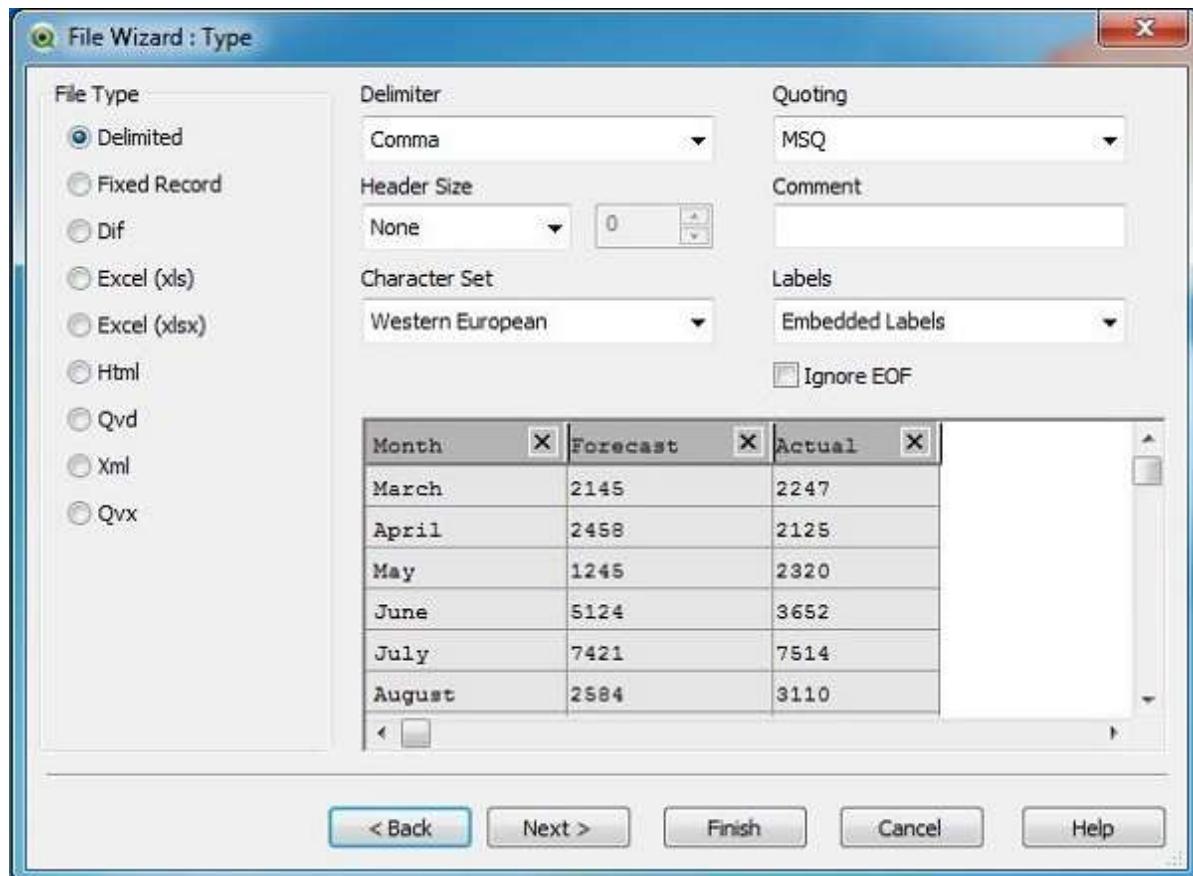
---

Let us consider the following input data, which represents the actual and forecasted sales figures.

Month	Forecast	Actual
March	2145	2247
April	2458,	
May	1245,	
June	5124	3652
July	7421	7514
August	2584,	
September	5314	4251
October	7846	6354
November	6532	7451
December	4625	1424
January	8547	7852
February	3265,	

## Load Script

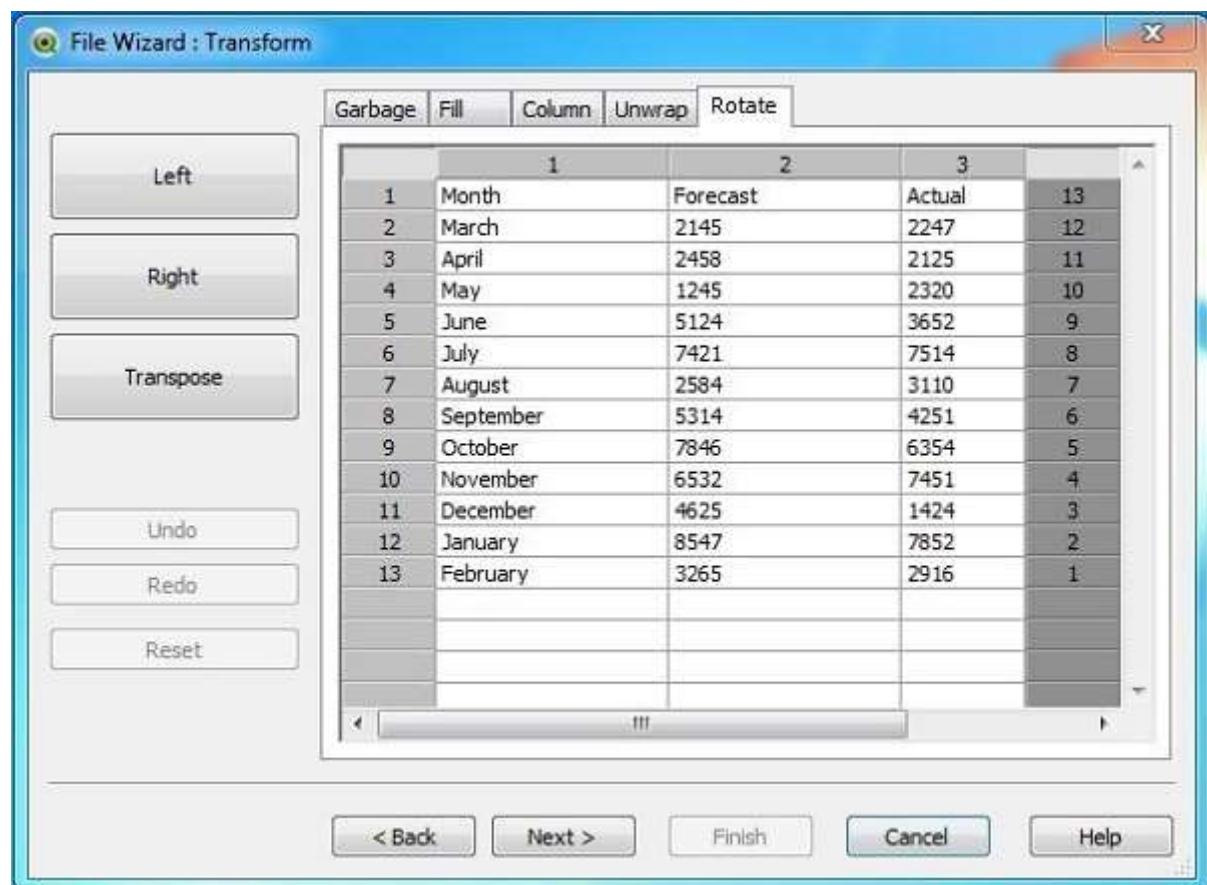
The above data is loaded to QlikView memory by using the script editor. Open the script editor from the File menu or press **Control+E**. Choose the "Table Files" option from the "Data from Files" tab and browse for the file containing the above data.



After clicking Next, we choose the **Enable Transformation Step** button to carry out the required data transformation.

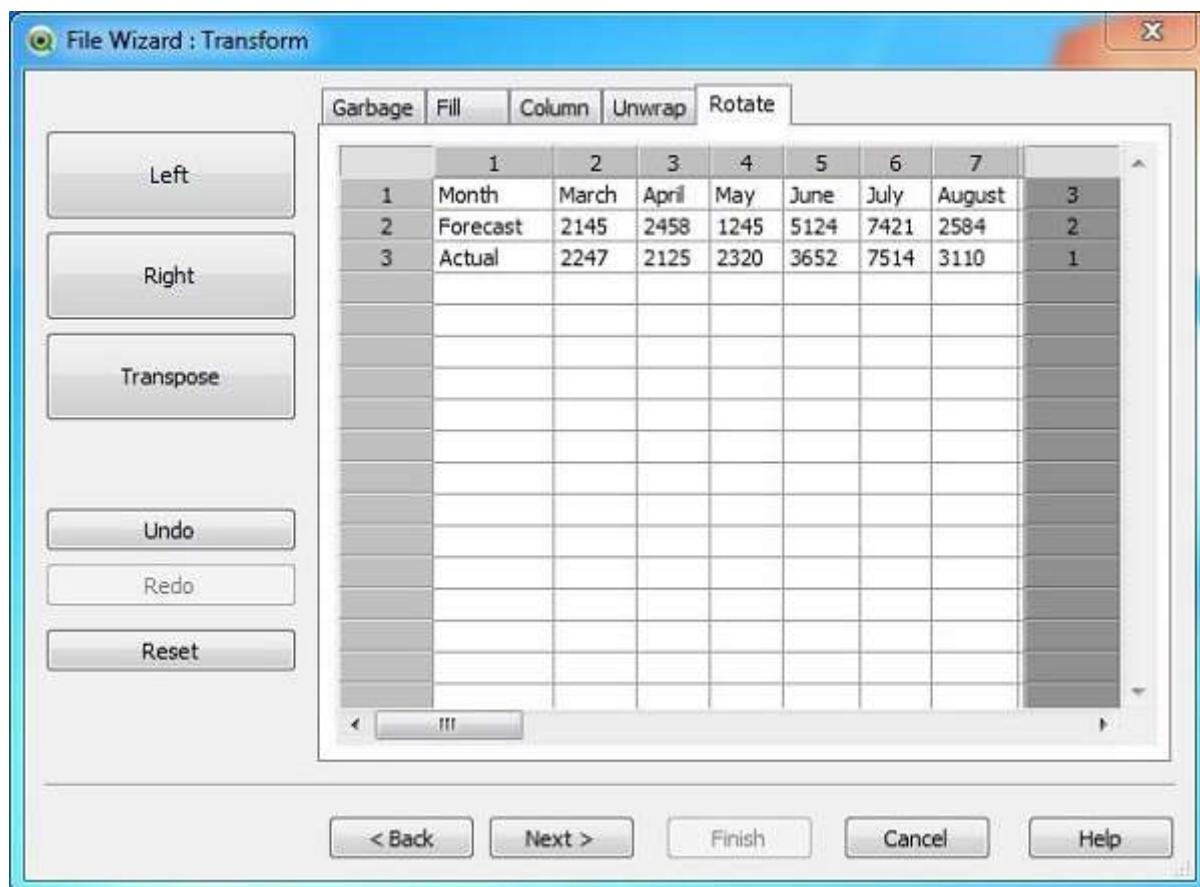
## Selecting the Data Transformation

As we are going to use the **Rotate** function, let us choose the Rotate tab which displays the values of all the fields.



## Apply Rotate

We click the **Transpose** button to transpose the above data. The transposed data appears as shown below.



## Load Script for transformed Data

The load script for the Transformed data can be seen using the script editor. The script shows the expression, which replaces the empty cell values.

The screenshot shows the 'Edit Script' dialog box in QlikView. The title bar reads 'Edit Script [C:\Qlikview\files\rotate.qvw]'. The menu bar includes File, Edit, Insert, Tab, Tools, Help, Reload, and Debug. The toolbar contains various icons for file operations like Open, Save, Print, and Find. The main area is titled 'Main' and contains the following QlikView script:

```
12 LOAD Month,
13   March,
14   April,
15   May,
16   June,
17   July,
18   August,
19   September,
20   October,
21   November,
22   December,
23   January,
24   February
25 FROM
26 [C:\Qlikview\data\quarterly_sales.csv]
27 (txt, codepage is 1252, embedded labels, delimiter is ',', msq, f
28 Transpose()
29 );
```

Below the script, there are tabs for Data, Functions, Variables, and Settings. The Data tab is selected. It contains settings for connecting to a database or files. Under 'Database', it says 'ODBC' with a dropdown arrow and a 'Connect...' button. There is also a checkbox for 'Force 32 Bit'. Under 'Data from Files', there are checkboxes for 'Relative Paths' and 'Use FTP', and buttons for 'Table Files...', 'QlikView File...', 'Web Files...', and 'Field Data...'. At the bottom of the dialog are 'OK', 'Cancel', and 'Help' buttons.

## Display Transformed Data

The transformed data can be seen by creating a **Table Box** using the option in the menu **Layout -> New Sheet Object**.

The screenshot shows the QlikView x64 Personal Edition interface. The title bar reads "QlikView x64 Personal Edition - [C:\Qlikview\files\rotate.qvw\*]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. The toolbar below the menu bar contains various icons for file operations, search, and navigation. A sub-menu titled "Main" is open. In the main workspace, there is a table box titled "Sales" containing data for different months. The table has two rows: "Actual" and "Forecast". The columns represent months from April to October. The data is as follows:

Month	April	August	December	February	January	July	June	March	May	November	October	...
Actual	2125	3110		1424	2916	7852	7514	3652	2247	2320	7451	6354
Forecast	2458	2584		4625	3265	8547	7421	5124	2145	1245	6532	7846

At the bottom of the screen, the status bar displays "For Help, press F1", the date and time "11/4/2015 7:20:38 AM\*", and the window size "2 X 13".

# QlikView Data Model

## 35. QlikView – Dimensions and Measures

Dimensions and Measures are fundamental entities, which are always used in data analysis. For example, consider the result of the analysis, "**what is the percentage change in volume of sales for each quarter?**" In this case, each quarter represents the **Dimensions**, which is the name of the quarter. The percentage change in volume represents the **Measures**, which is a calculation with respect to each value in the dimension. Below are some widely accepted definition of these two terms.

- **Dimension:** It is a descriptive field in the data set, which represents few distinct values. Examples - Month, Year, Product ID etc.
- **Measures:** It is a numeric field on which some calculations are performed for each distinct value of dimension.

### Input Data

---

Let us consider the following input data, which represents the sales volume and Revenue of different product lines and product categories in different regions. Save the data into a .csv file.

```
ProductID,ProductCategory,Region,SalesVolume, Revenue
1,Outdoor Recreation,Europe,457,25841
2,Clothing,Europe,125,54281
3,Costumes & Accessories,South Asia,781,54872
4,Athletics,South Asia,839,87361
5,Personal Care,Australia,473,15425
6,Arts & Entertainment,North America,625,84151
7,Hardware,South America,772,45812
```

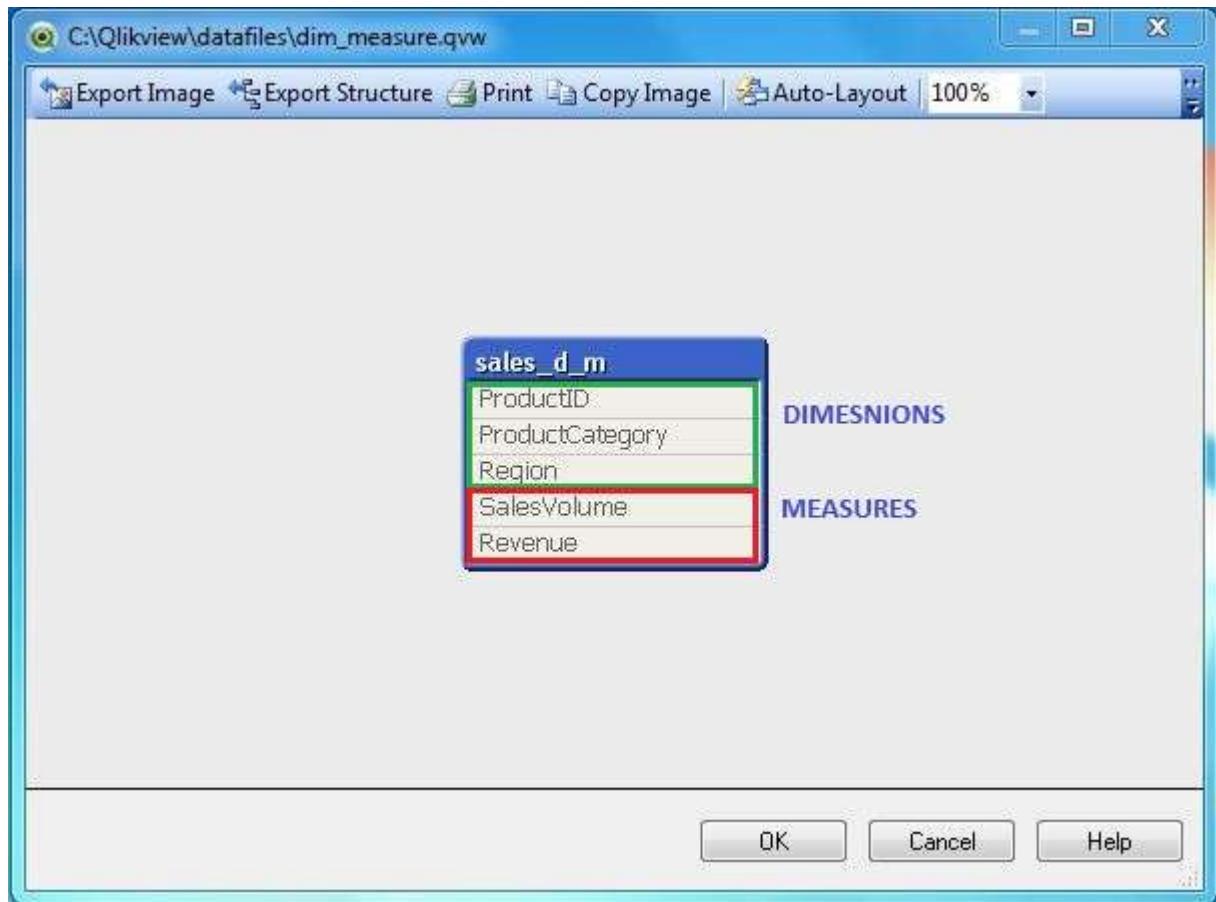
### Load Script

---

The above data is loaded to the QlikView memory by using the script editor. Open the Script editor from the File menu or press **Control+E**. Choose the **Table Files** option from the **Data from Files** tab and browse for the file containing the above data. Click **OK** and press **Control+R** to load the data into the QlikView's memory.

## Table Structure

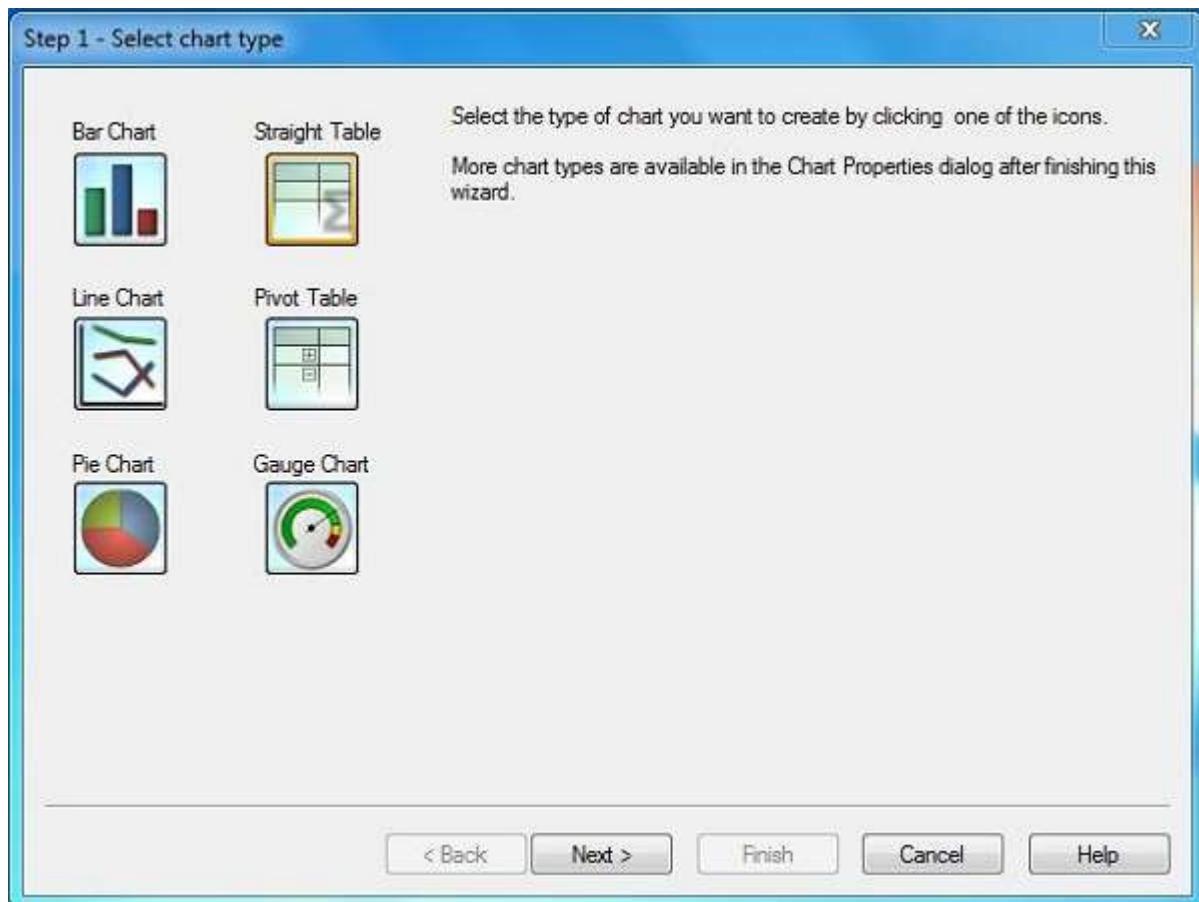
We can see the structure of the table by following the menu **File -> Table Viewer** or pressing **Control+T**. The following screen comes up in which we have marked the dimensions inside a green box and the measures inside a red box.



## Using Dimensions and Measures

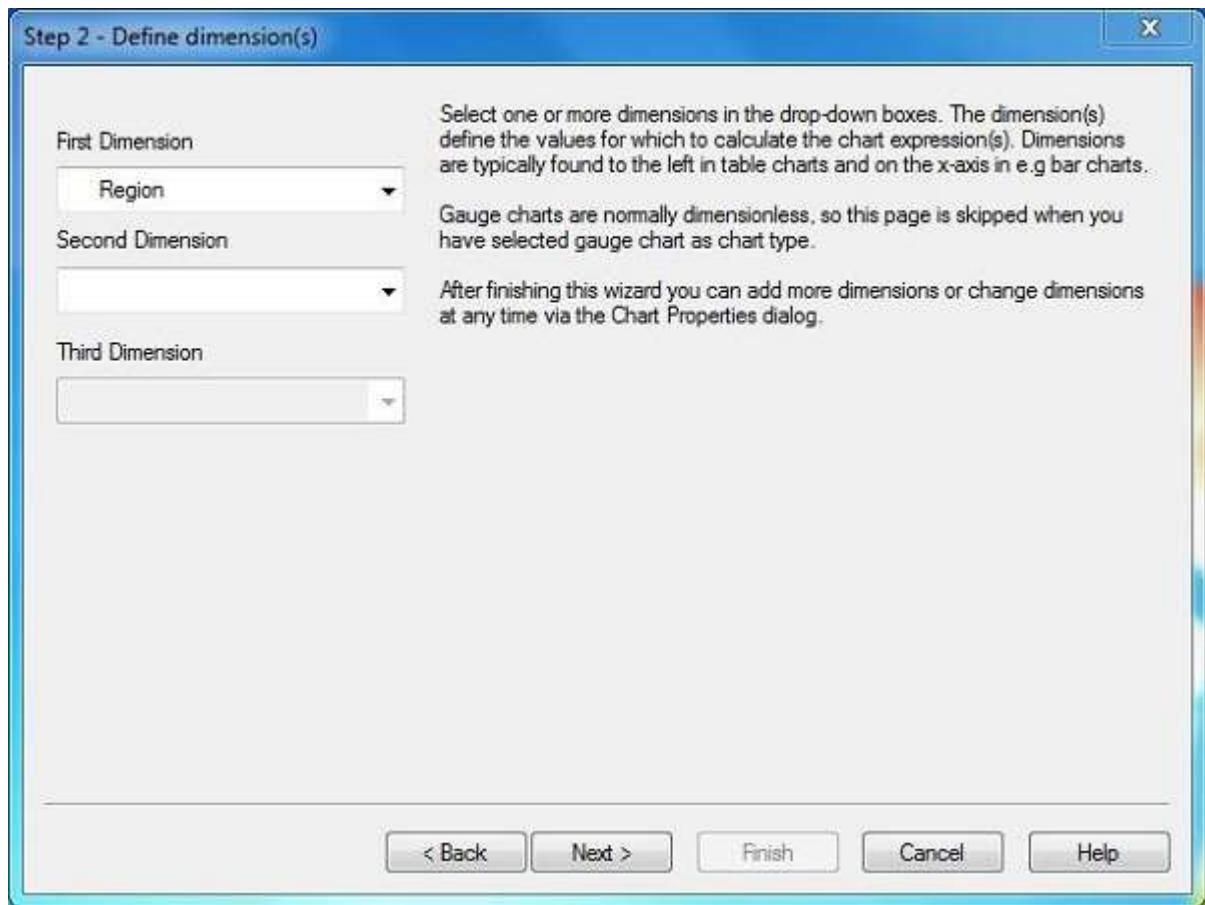
Let us create a straight table chart showing the calculation using above dimensions and measures. Click on the **Quick Chart Wizard** as shown below.

Next, click on the **Straight Table** option. Click Next.



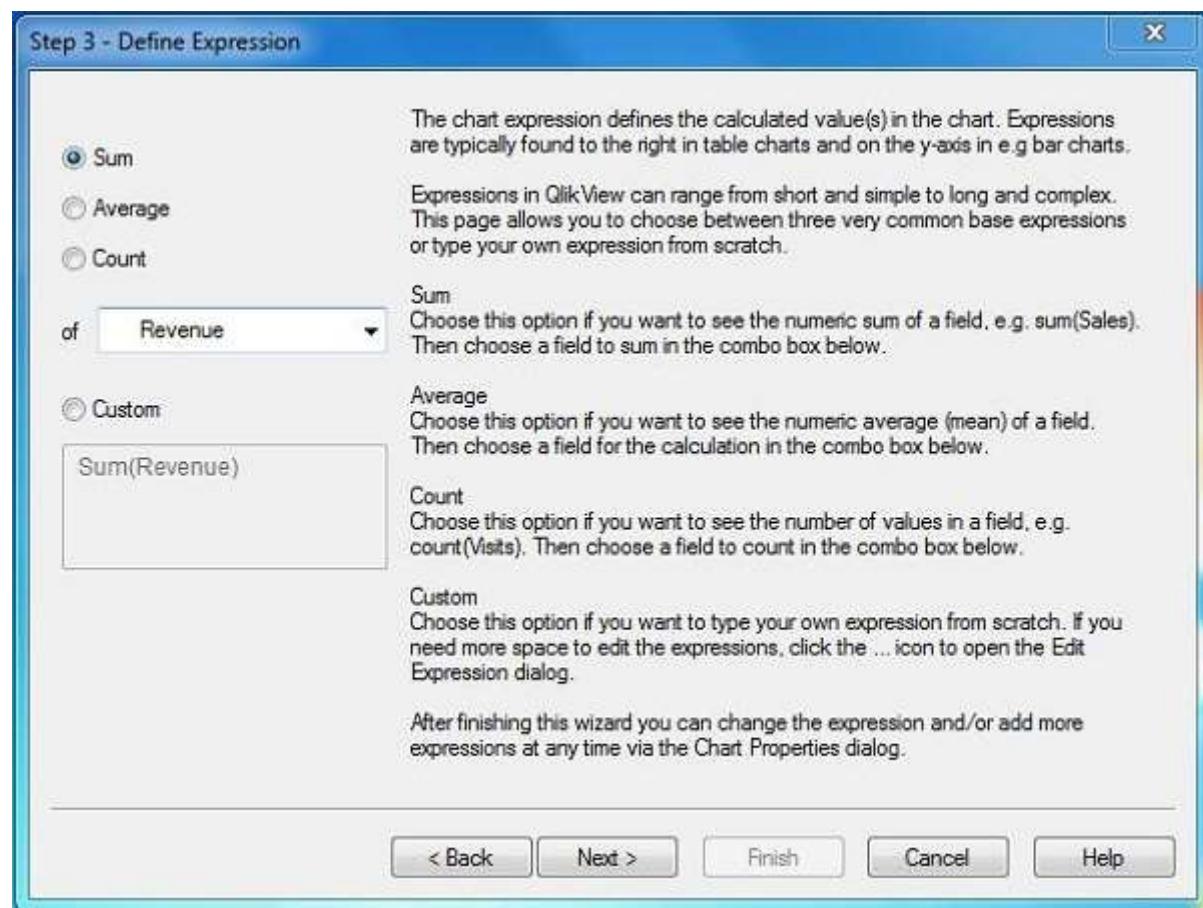
## Choose Dimensions

In this screen, we choose Region as the dimension as we want to select the total revenue for each region.



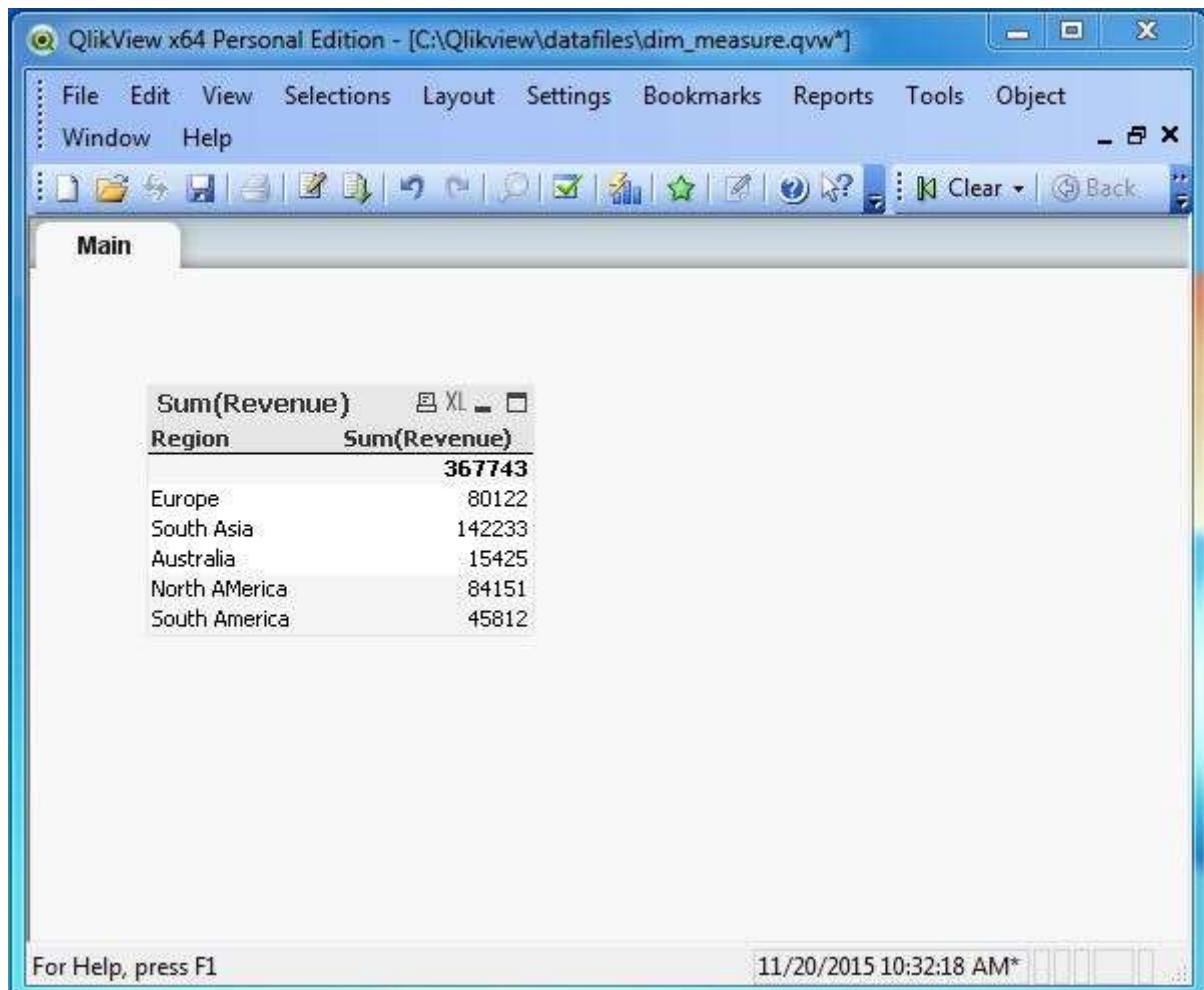
## Choose Measure

The Next screen prompts for applying the calculation on a measure field. We choose to apply **Sum** on the field Revenue.



## Final chart

On completing the above steps, we get the final chart which shows the total revenue(Measure) for each region(Dimension).



The screenshot shows the QlikView interface with a table titled "Sum(Revenue)" displayed. The table has two columns: "Region" and "Sum(Revenue)". The total value for all regions is 367743.

Region	Sum(Revenue)
Europe	80122
South Asia	142233
Australia	15425
North AMerica	84151
South America	45812

For Help, press F1      11/20/2015 10:32:18 AM\*

## 36. QlikView – Star Schema

A star schema model is a type of data model in which multiple dimensions are linked to a single fact table. Of course, in bigger models there can be multiple facts tables linked to multiple dimensions and other fact tables. The usefulness of this model lies in performing fast queries with minimal joins among various tables. The fact table contains data, which are measures and have numeric values. Calculations are applied on the fields in the fact table. The unique keys of the dimension tables are used in linking it to the fat table, which also has a key usually with the same field name. Therefore, the Fact table contains the keys from the entire dimension table and forms a concatenated primary key used in various queries.

### **Input Data**

---

Given below is a list of tables, which contain the data for different products from various suppliers and regions. Also the supply happens at different time intervals, which are captured in the Time dimension table.

#### **Product Dimension**

It contains the Product Category and Product Names. The Product ID field is the unique Key.

```
ProductID,ProductCategory,ProductName
1,Outdoor Recreation,Winter Sports & Activities
2,Clothing,Uniforms
3,Lawn & Garden Power, Equipment
4,Athletics,Rugby
5,Personal Care,Shaver
6,Arts & Entertainment,Crafting Materials
7,Hardware,Power Tool Batteries
```

#### **Region Dimension**

It contains the Region Names where the suppliers are based. The RegionID field is the unique Key.

```
RegionID,Continent,Country
3,North America, USA
7,South America, Brazil
12,Asia,China
2,Asia,Japan
5,Europe,Belgium
```

## Supplier Dimension

It contains the Supplier Names, which supply the above products. The SupplierID field is the unique Key.

SupplierID,SupplierName
3S12,Supre Suppliers
4A15,ABC Suppliers
4S66,Max Sports
5F244,Nice Foods
8A45,Artistic angle

## Time Dimension

It contains the Time periods when the supply of the above products occur. The TimeID field is the unique Key.

TimeID,Year,Month
1,2012,Feb
2,2012,May
3,2012,Sep
4,2013,Aug
5,2014,Jan
6,2014,Nov

## Supplier Quantity Fact

It contains the values for the quantities supplied and percentage of defects in them. It joins to each of the above dimensions through keys with same name.

ProductID,RegionID,TimeID,SupplierID,Quantity, DefectPercentage
1,3,3,5F244,8452,12
2,3,1,4S66,5124,8.25
3,7,1,8A45,5841,7.66
4,12,2,4A15,5123,1.25
5,5,3,4S66,7452,8.11
6,2,5,4A15,5142,3.66
7,2,1,4S66,452,2.06

## Load Script

The above data is loaded to QlikView memory by using the script editor. Open the Script editor from the File menu or press **Control+E**. Choose the **Table Files** option from the **Data from Files** tab and browse for the file containing the above data. Click **OK** and

205

press **Control+R** to load the data into QlikView's memory. Below is the script which appears after each of the above file is read.

```
LOAD ProductID,
    ProductCategory,
    ProductName
FROM
[C:\Qlikview\images\StarSchema\Product_dimension.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);

LOAD TimeID,
    Year,
    Month
FROM
[C:\Qlikview\images\StarSchema\Time.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);

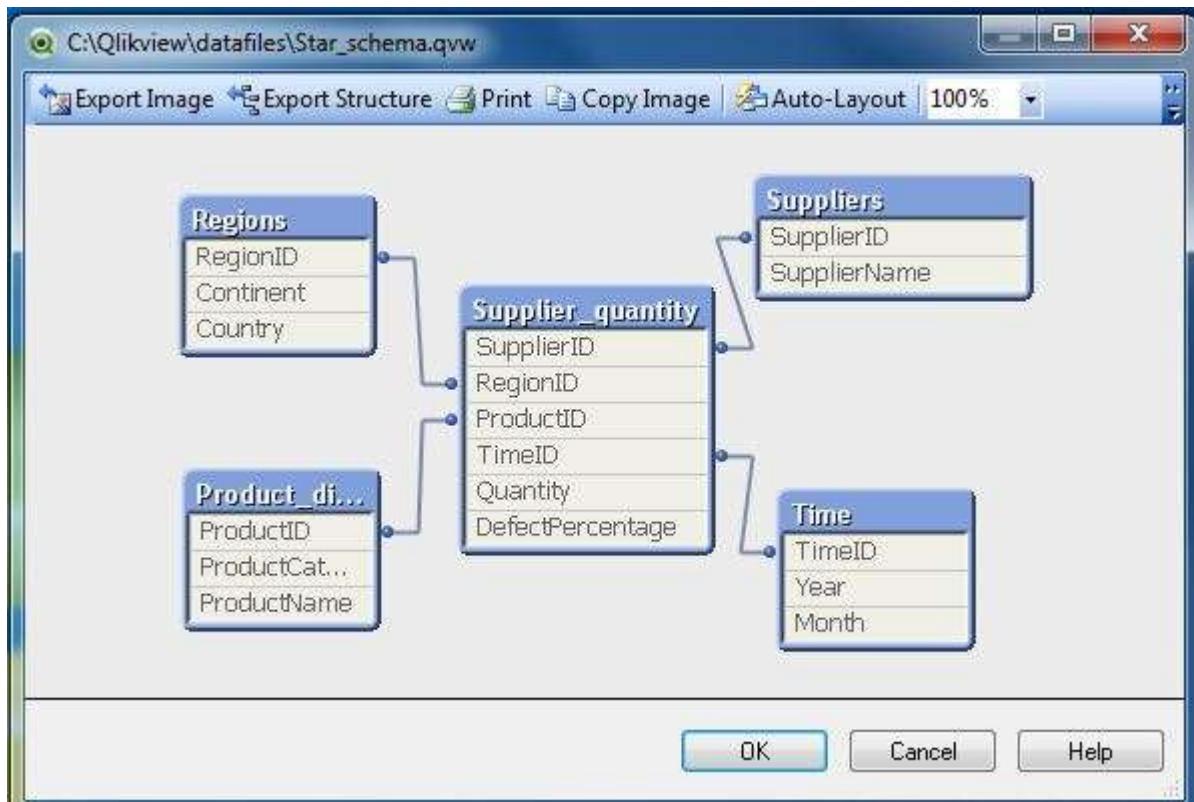
LOAD SupplierID,
    SupplierName
FROM
[C:\Qlikview\images\StarSchema\Suppliers.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);

LOAD RegionID,
    Continent,
    Country
FROM
[C:\Qlikview\images\StarSchema\Regions.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);

LOAD ProductID,
    RegionID,
    TimeID,
    SupplierID,
    Quantity,
    DefectPercentage
FROM
[C:\Qlikview\images\StarSchema\Supplier_quantity.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);
```

## Star Schema Data Model

After reading the above data into QlikView memory, we can look at the data model, which shows all the tables, fields, and relationship in form of a star schema.



## 37. QlikView – Synthetic Key

A **Synthetic Key** is QlikView's solution to create an artificial key when there is ambiguity about which key to use between two tables. This situation arises when two tables have two or more fields in common. QlikView's feature of creating association in memory automatically detects this scenario and creates an additional table, which will hold the value of the new key created.

### Input Data

---

Let us consider the following two CSV data files, which are used as input for further illustrations.

Sales:

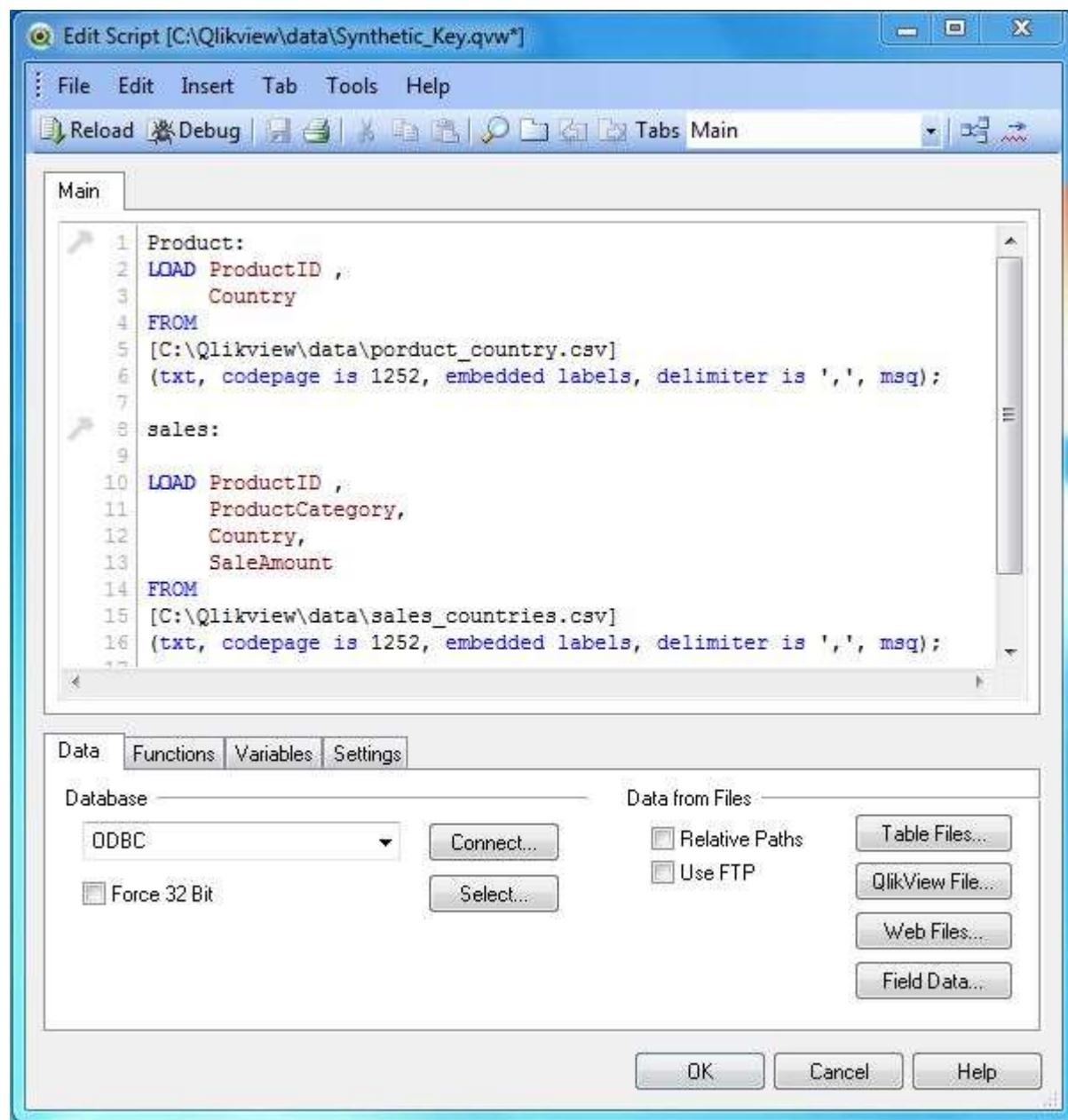
```
ProductID,ProductCategory,Country,SaleAmount  
1,Outdoor Recreation,Italy,4579  
2,Clothing,USA,4125  
3,Costumes & Accessories,South Korea,6521
```

Product:

```
ProductID, Country  
3,Brazil  
3,China  
2,Korea  
1,USA
```

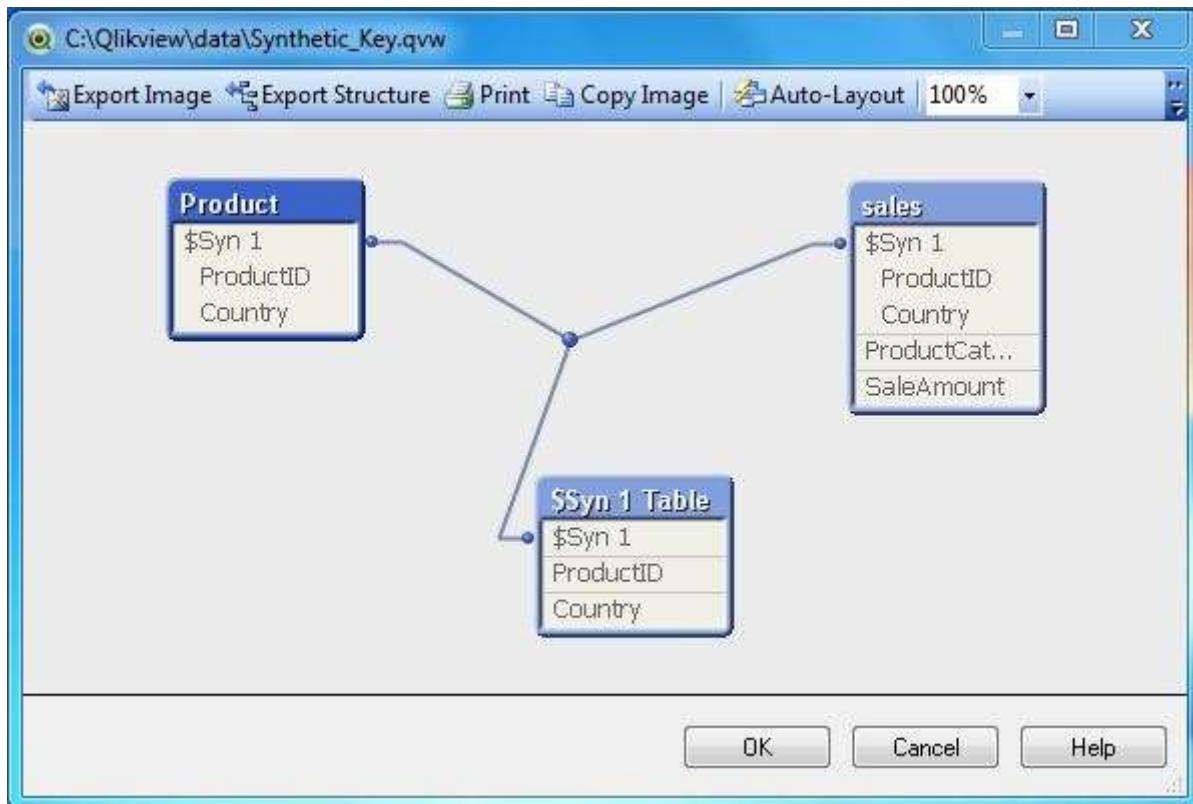
## Load Script

We load the above input data using the script editor, which is invoked by pressing **Control+E**. Choose the option **Table Files** and browse for the Input file.



## Data Model

Next, we look at the data model by using the menu command for table viewer, **Control+T**. The following screen comes up, which shows the creation of a third table that supplies the value of the synthetic key as both the tables have ProductID and Country as matching keys.



## Impact of Synthetic key

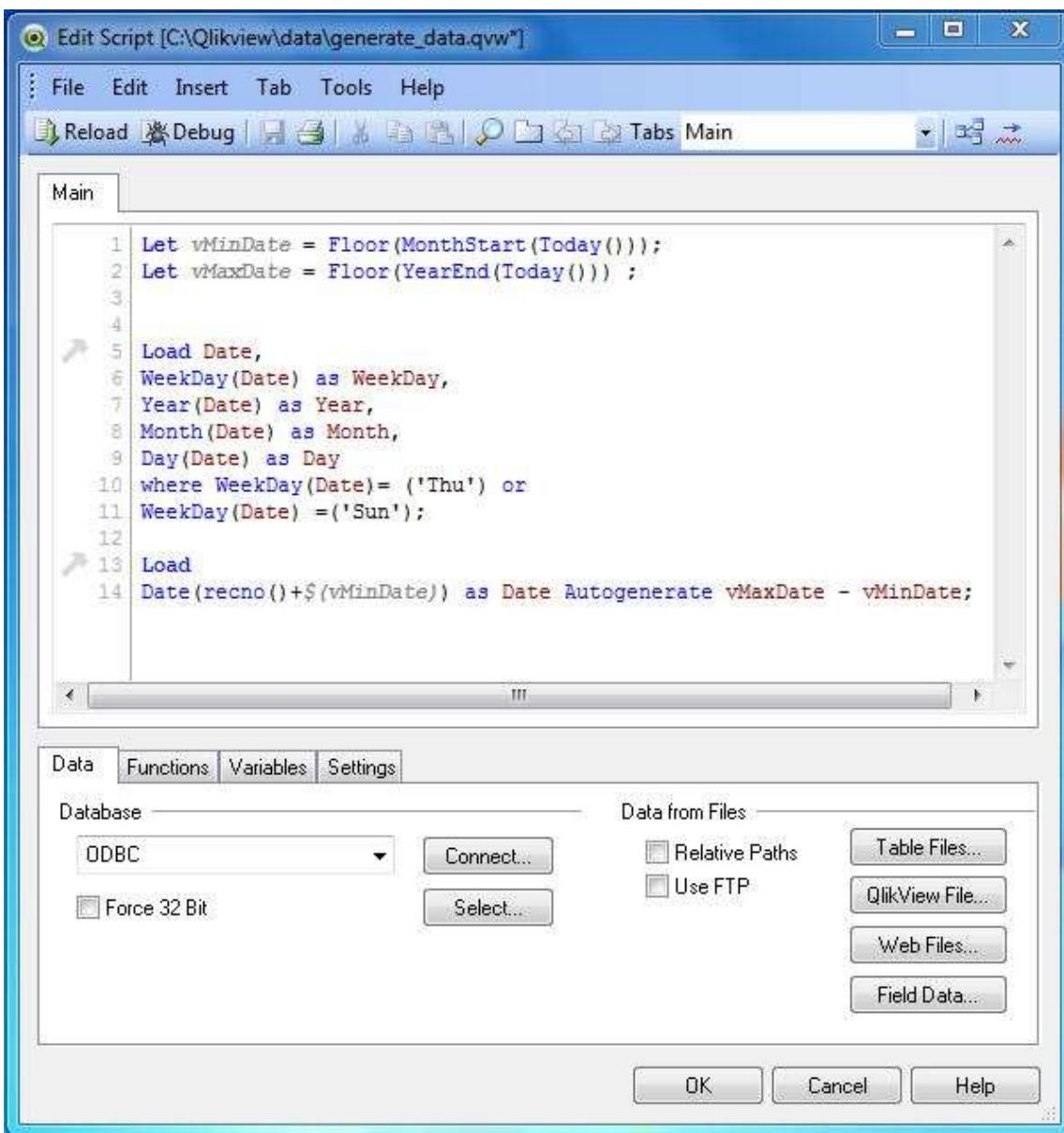
Synthetic keys indicate the flaw in the data model that is being used. They do not cause any issue in the correctness of the data or performance of the report. Things will work fine if a big data model has one or two instances of synthetic keys. However, if we have too many of them, then that is an implication to redesign the data model.

## 38. QlikView – Generating Data

Many times, we need some data to be generated programmatically by the software being used, which is not coming from a source. For example, 100 random numbers or just the dates of 23rd week of a year. A data analyst may need such data to be created to perform some analysis on the data that does not contain these values as it arrived. QlikView provides a function called **Autogenerate**, which can be used for such requirement.

### Data Requirement

Consider a scenario where we need to find only the dates, which are a Thursday or a Sunday. We need to find it for the range starting today until the end of the year. We create the following script, which will achieve this.



The screenshot shows the 'Edit Script' dialog box in QlikView. The title bar says 'Edit Script [C:\Qlikview\data\generate\_data.qvw]'. The menu bar includes File, Edit, Insert, Tab, Tools, Help. The toolbar has icons for Reload, Debug, and various file operations. The main area is titled 'Main' and contains the following QVW script:

```
Let vMinDate = Floor(MonthStart(Today()));
Let vMaxDate = Floor(YearEnd(Today()));

Load Date,
WeekDay(Date) as WeekDay,
Year(Date) as Year,
Month(Date) as Month,
Day(Date) as Day
where WeekDay(Date)= ('Thu') or
WeekDay(Date) =('Sun');

Load
Date(recno()+$ (vMinDate)) as Date Autogenerate vMaxDate - vMinDate;
```

Below the script, there are tabs for Data, Functions, Variables, and Settings. Under the Data tab, there are sections for Database (ODBC, Force 32 Bit, Connect..., Select...) and Data from Files (Relative Paths, Use FTP, Table Files..., QlikView File..., Web Files..., Field Data...). At the bottom are OK, Cancel, and Help buttons.

## Explanation

We declare two variables to capture the first day of the current month and end of the year. Next, we apply various functions and a filter condition to generate the required values. The recno() function creates one record for each of these dates. We add Autogenerate function giving the variables as the range.

## Generated data

On loading the above script to QlikView's memory and creating a **Table Box** using the menu **Layout -> New Sheet Objects -> Table Box**, we get the data created as shown below.

The screenshot shows the QlikView interface with a window titled "QlikView x64 Personal Edition - [C:\Qlikview\data\generate\_data.qvw]". The main area is titled "Main" and contains a "Calendar" table box. The table has columns: Date, WeekDay, Day, Month, and Year. The data is as follows:

Date	WeekDay	Day	Month	Year
11/5/2015	Thu	5	Nov	2015
11/8/2015	Sun	8	Nov	2015
11/12/2015	Thu	12	Nov	2015
11/15/2015	Sun	15	Nov	2015
11/19/2015	Thu	19	Nov	2015
11/22/2015	Sun	22	Nov	2015
11/26/2015	Thu	26	Nov	2015
11/29/2015	Sun	29	Nov	2015
12/3/2015	Thu	3	Dec	2015
12/6/2015	Sun	6	Dec	2015
12/10/2015	Thu	10	Dec	2015
12/13/2015	Sun	13	Dec	2015
12/17/2015	Thu	17	Dec	2015
12/20/2015	Sun	20	Dec	2015
12/24/2015	Thu	24	Dec	2015
12/27/2015	Sun	27	Dec	2015
12/31/2015	Thu	31	Dec	2015

At the bottom of the screen, there is a status bar with the text "For Help, press F1", the date and time "11/22/2015 5:07:12 PM", and the size "17 X 5".

# QlikView Data Analysis

## 39. QlikView – Cross Tables

While analyzing data, we come across situations where we desire columns to become rows and vice-versa. It is not just about transposing, it also involves rolling up many columns together or repeating many values in a row many times to achieve the desired column and row layout in the table.

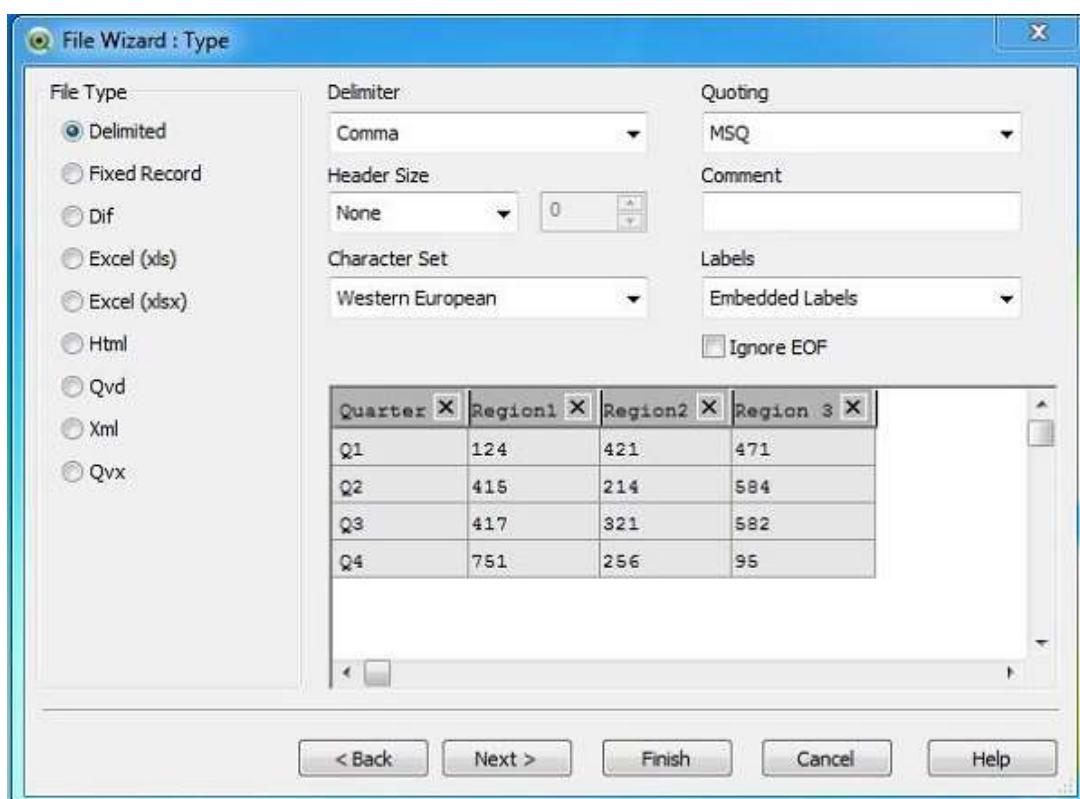
### Input data

Consider the following input data, which shows region wise sales of a certain product for each quarter. We create a delimited file (CSV) with the below given data.

```
Quarter,Region1,Region2,Region 3
Q1,124,421,471
Q2,415,214,584
Q3,417,321,582
Q4,751,256,95
```

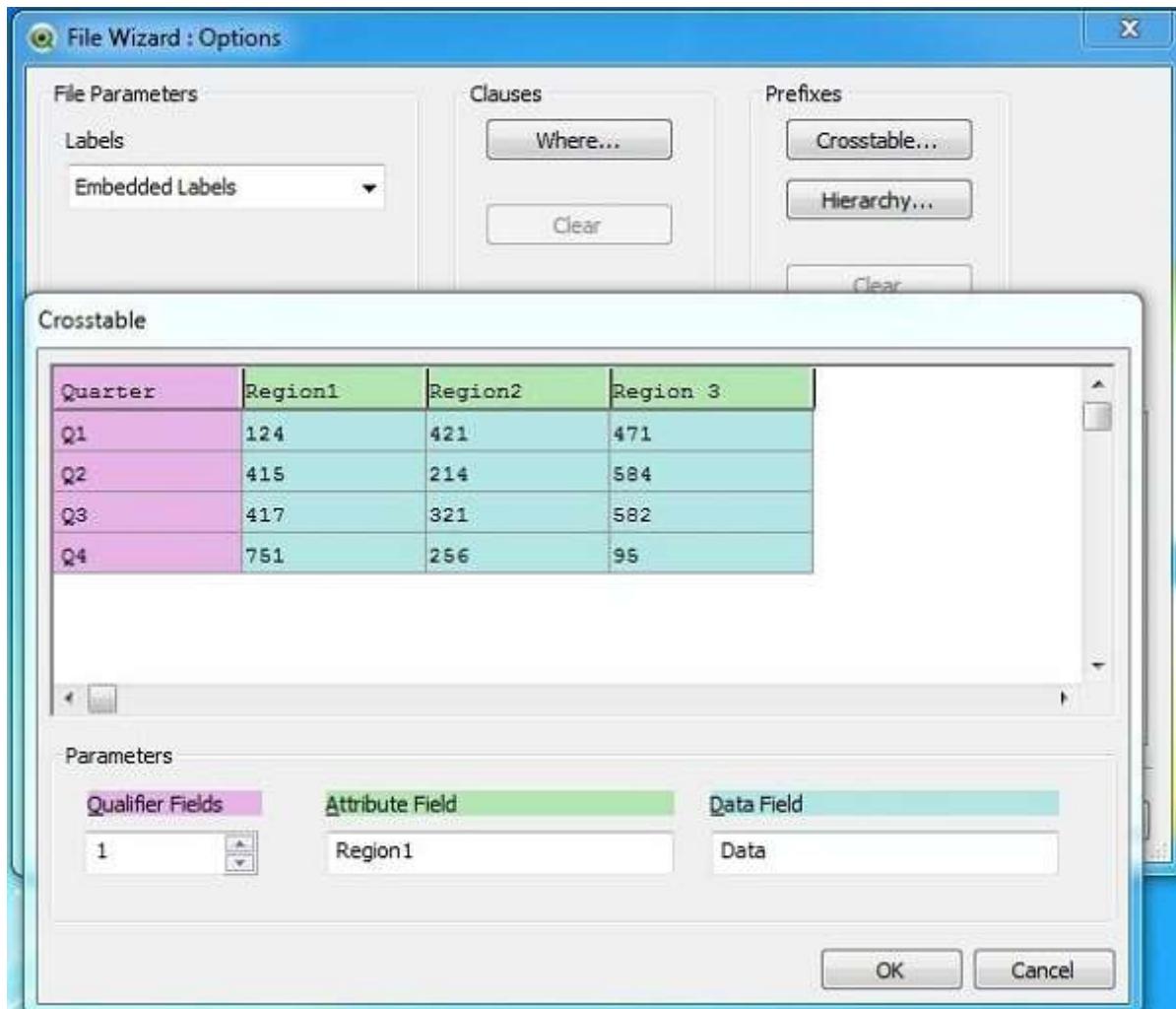
### Loading Input Data

We load the above input data using the script editor, which is invoked by pressing **Control+E**. Choose the option **Table Files** and browse for the Input file. After choosing the options as shown below, click Next.



## Crosstable Options

In the next window (File Wizard -> Options), click on the **Crosstable** button. It highlights the columns in different colors. The pink color shows the **qualifier field**, which is going to be repeated across many rows for each value of in the **Attribute Field**. The cell values under the Attribute fields are taken as the data. Click OK.



## Crosstable Transformation

The transformed data appears in which all the Region fields are clubbed to one column but with values repeating for each quarter.

The screenshot shows the 'File Wizard : Options' dialog box. In the 'File Parameters' section, the 'Labels' dropdown is set to 'Embedded Labels'. In the 'Clauses' section, there are 'Where...' and 'Clear' buttons. In the 'Prefixes' section, the 'Crosstable...' button is highlighted in blue, while 'Hierarchy...' and another 'Clear' button are grayed out. Below the dialog, a preview window displays a table titled 'Result' with three columns: Quarter, Region1, and Data. The data is as follows:

Quarter	Region1	Data
Q1	Region1	124
Q1	Region2	421
Q1	Region 3	471
Q2	Region1	415
Q2	Region2	214
Q2	Region 3	584
Q3	Region1	417
Q3	Region2	321
Q3	Region 3	582
Q4	Region1	751
Q4	Region2	256
Q4	Region 3	95

At the bottom of the dialog, there are buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

## Load Script

The Load script for the crosstable transformations shows the commands given below.

The screenshot shows the 'Edit Script [QV3]' dialog box. The main pane displays the following QlikView script:

```
12 CrossTable(Region1, Data)
13 LOAD Quarter,
14     Region1,
15     Region2,
16     [Region 3]
17 FROM
18 [C:\Qlikview\data\quarterly_sales_forecast.csv]
19 (txt, codepage is 1252, embedded labels, delimiter is ',', msq);
```

The 'Main' tab is selected. Below the script editor, there are tabs for 'Data', 'Functions', 'Variables', and 'Settings'. The 'Data' tab is active. Under 'Database', the dropdown is set to 'ODBC' and the checkbox 'Force 32 Bit' is checked. Under 'Data from Files', there are checkboxes for 'Relative Paths' and 'Use FTP', and buttons for 'Table Files...', 'QlikView File...', 'Web Files...', and 'Field Data...'. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

## Crosstable Data

On creating a Table Box sheet object using the menu **Layout -> New Sheet Objects -> Table Box**, we get the following output.

The screenshot shows the QlikView x64 Personal Edition interface. The title bar reads "QlikView x64 Personal Edition - [C:\Qlikview\files\cross\_tab.qvw\*]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. The toolbar below the menu bar contains various icons for file operations like Open, Save, Print, and search. The main workspace is titled "Main" and displays a crosstable titled "Quaterly Sales". The table has three columns: "Quarter", "Region1", and "Data". The data rows are as follows:

Quarter	Region1	Data
Q1	Region1	124
Q1	Region2	421
Q1	Region 3	471
Q2	Region2	214
Q2	Region1	415
Q2	Region 3	584
Q3	Region2	321
Q3	Region1	417
Q3	Region 3	582
Q4	Region 3	95
Q4	Region2	256
Q4	Region1	751

At the bottom left of the workspace, it says "For Help, press F1". At the bottom right, it shows the date and time "11/4/2015 9:32:16 AM\*" and the dimensions "12 X 3".

## 40. QlikView – Straight Tables

Straight Tables are most widely used sheet object to display data in QlikView. They are very simple yet powerful with features like column rearrangement, sorting and coloring the background etc.

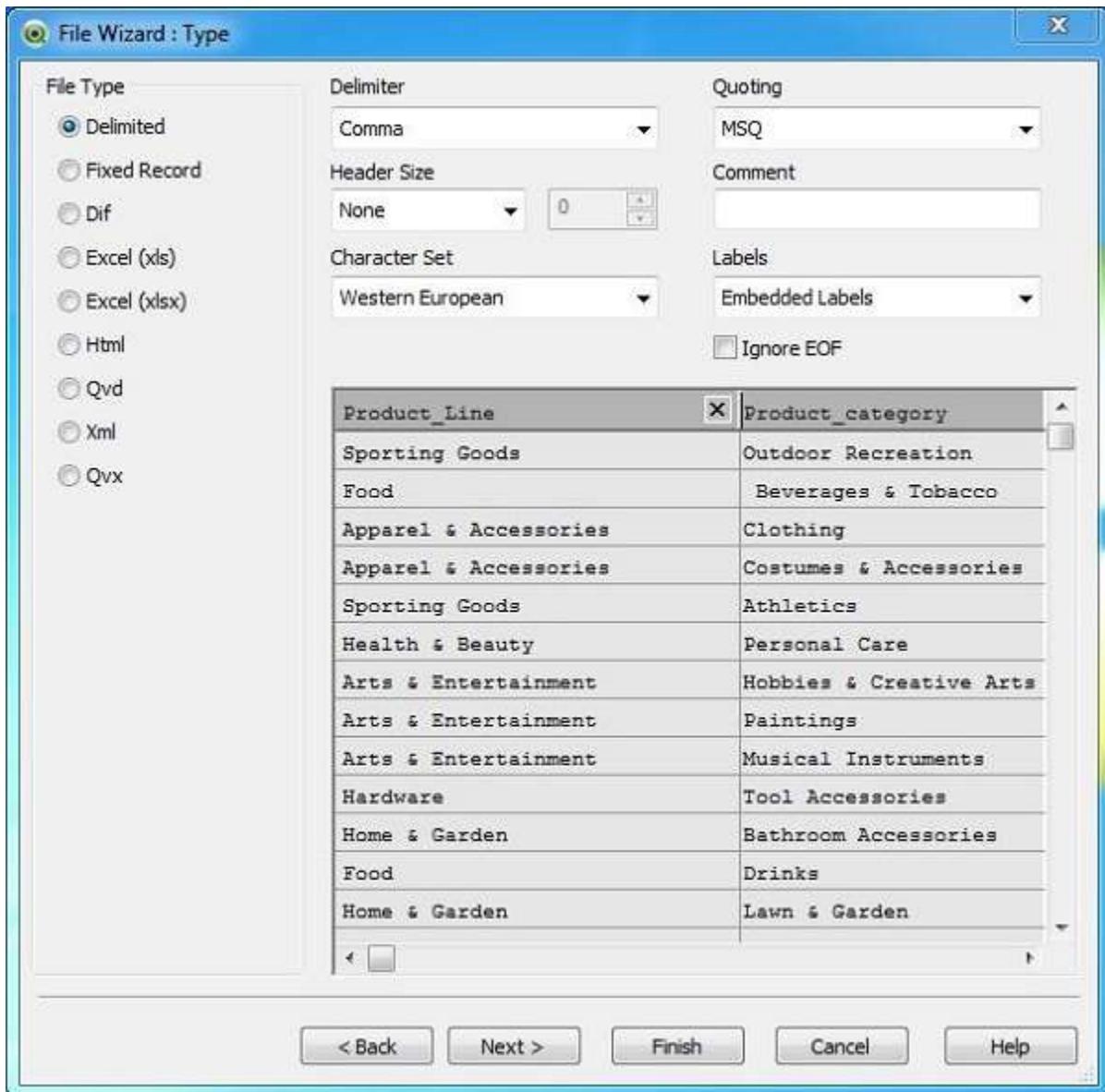
### Input Data

Let us consider the following input data, which represents the sales figure of different product lines and product categories.

Product_Line	Product_category	Value
Sporting Goods	Outdoor Recreation	5642
Food	Beverages & Tobacco	2514
Apparel & Accessories	Clothing	2365
Apparel & Accessories	Costumes & Accessories	4487
Sporting Goods	Athletics	812
Health & Beauty	Personal Care	6912
Arts & Entertainment	Hobbies & Creative Arts	5201
Arts & Entertainment	Paintings	8451
Arts & Entertainment	Musical Instruments	1245
Hardware	Tool Accessories	456
Home & Garden	Bathroom Accessories	241
Food	Drinks	1247
Home & Garden	Lawn & Garden	5462
Office Supplies	Presentation Supplies	577
Hardware	Blocks	548
Baby & Toddler	Diapering	1247
Baby & Toddler	Toys	257
Home & Garden	Pipes	1241
Office Supplies	Dispaly Borad	2177

## Load Script

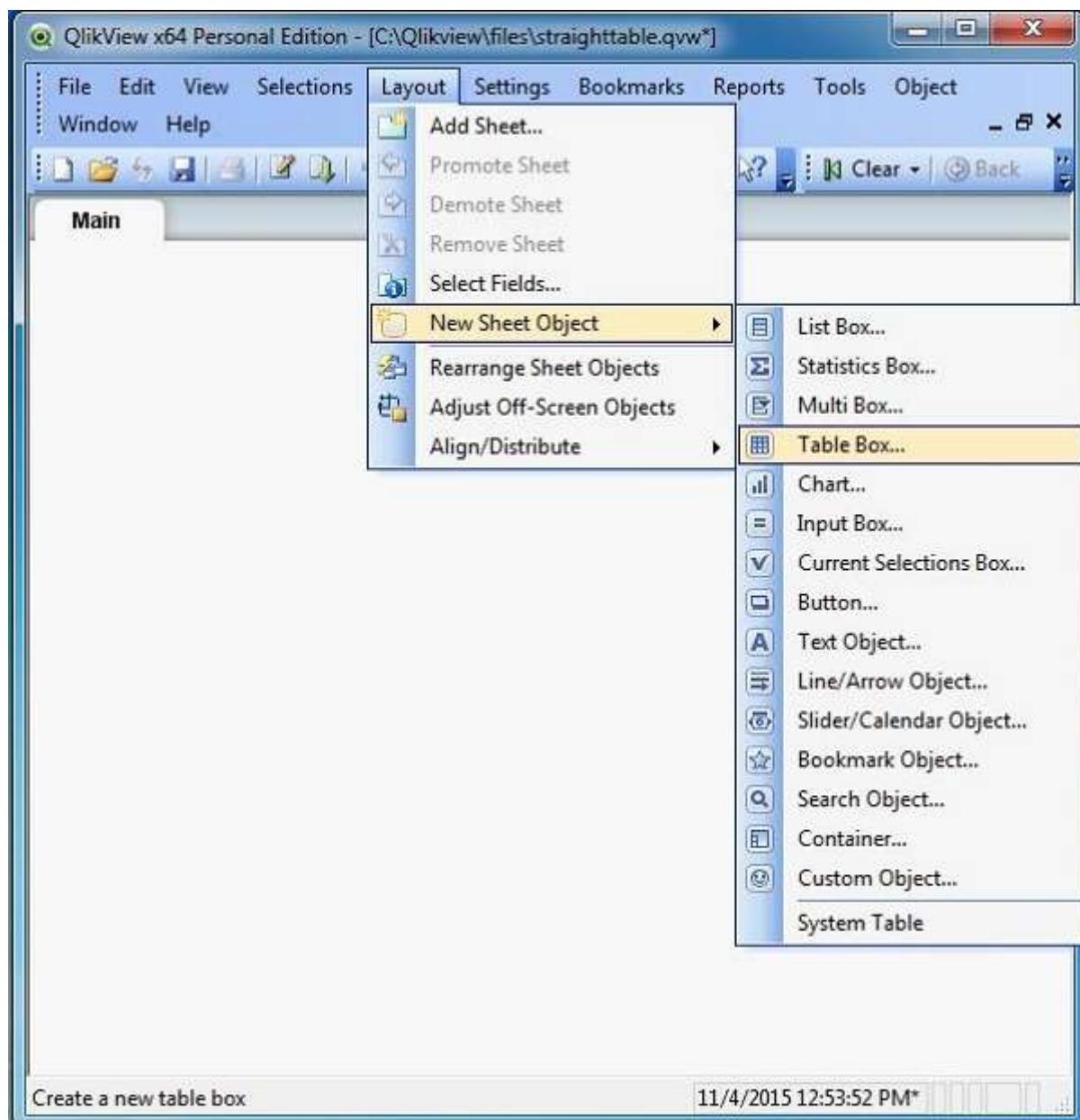
The above data is loaded to the QlikView memory by using the script editor. Open the Script editor from the File menu or press **Control+E**. Choose the "Table Files" option from the "Data from Files" tab and browse for the file containing the above data. The following screen appears.



Click "OK" and press "Control+R" to load the data into the QlikView's memory.

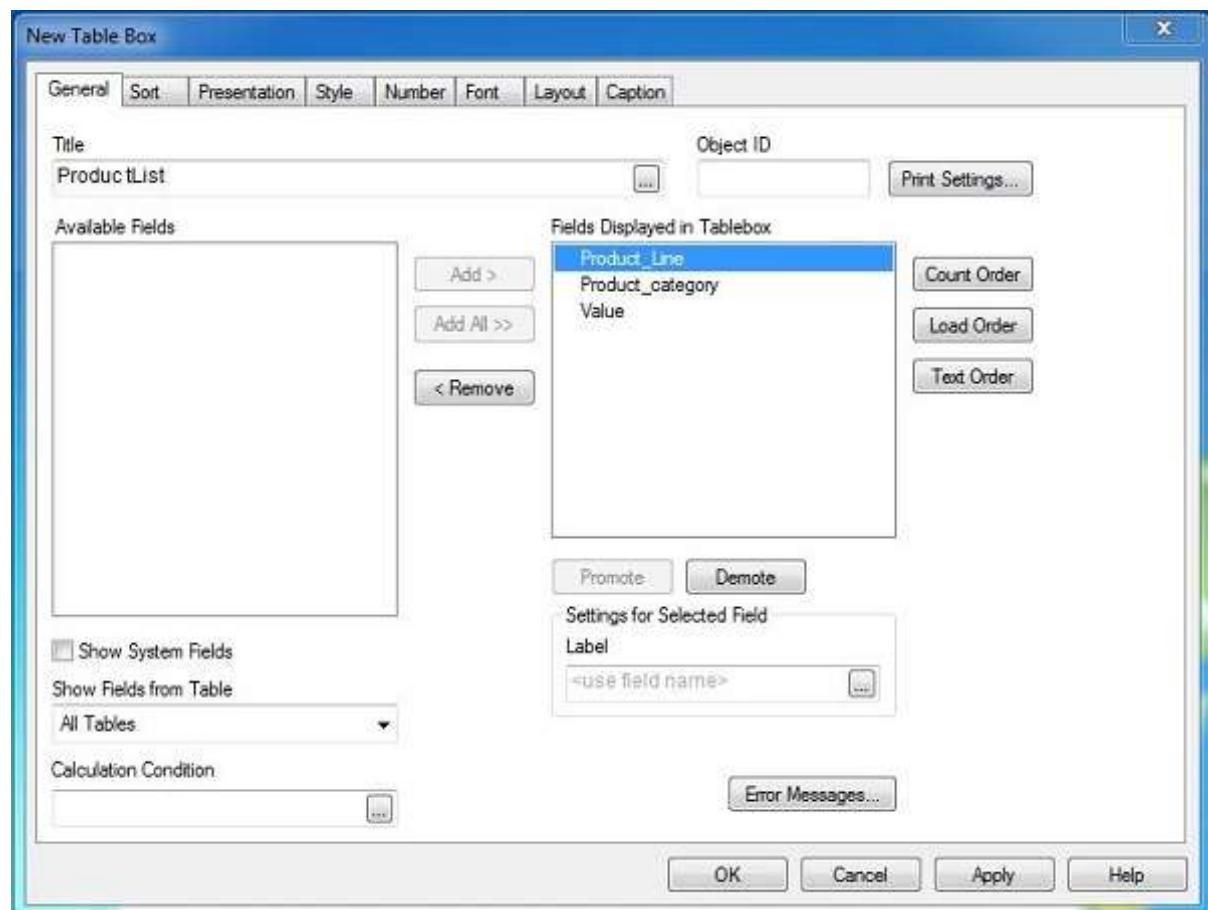
## Create Table Box

Next, we create a new sheet Object of type Table Box. We follow the menu as shown below.



## Select the Columns

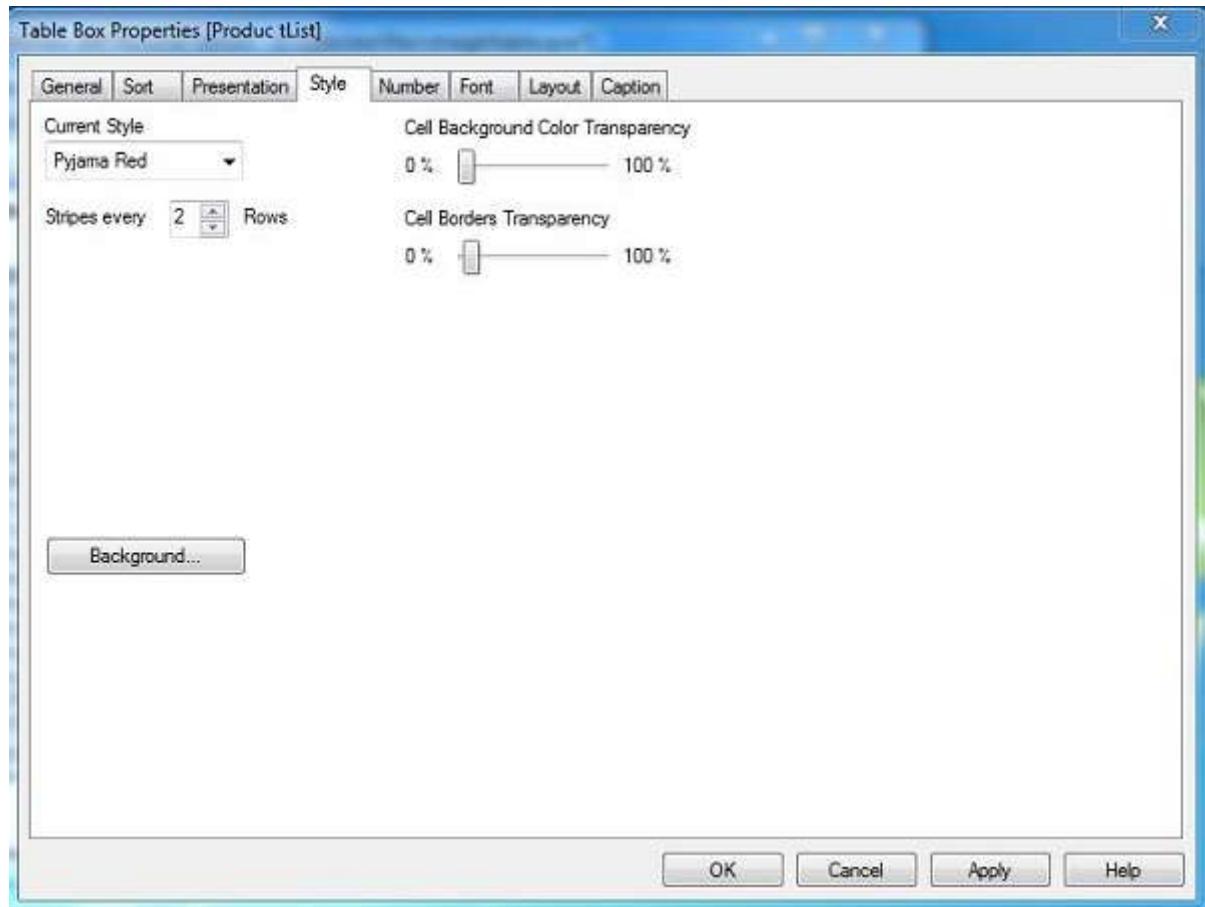
QlikView prompts for the columns to be chosen which will be displayed in the final Table Box. We choose all the columns and use the **Promote or Demote** option to set the order of the columns.



## Select Display Style

222

Next, we choose the style tab to give specific background colors to the display data. The current style option lists many pre-built styles. We choose **Pyjama Red** with **Stripes** every two rows.



## Column Re-ordering

223

We can reorder the positions of the columns by pressing and holding the mouse button at the column headers and then dragging it to the desired position.

The screenshot shows the QlikView interface with a window titled "QlikView x64 Personal Edition - [C:\Qlikview\files\straighttable.qvw\*]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. Below the menu is a toolbar with various icons. The main area is titled "Main" and contains a table titled "Product tList". The table has three columns: "Product\_Line", "Product\_category", and "value". The "value" column header is highlighted with a blue arrow pointing to its top edge. The data in the table is as follows:

Product_Line	Product_category	value
Apparel & Accessories	Clothing	2365
Apparel & Accessories	Costumes & Accessories	4487
Arts & Entertainment	Hobbies & Creative Arts	5201
Arts & Entertainment	Musical Instruments	1245
Arts & Entertainment	Paintings	8451
Baby & Toddler	Diapering	1247
Baby & Toddler	Toys	257
Food	Beverages & Tobacco	2514
Food	Drinks	1247
Hardware	Blocks	548
Hardware	Tool Accessories	456
Health & Beauty	Personal Care	6912
Home & Garden	Bathroom Accessories	241
Home & Garden	Lawn & Garden	5462
Home & Garden	Pipes	1241
Office Supplies	Display Board	2177
Office Supplies	Presentation Supplies	537
Sporting Goods	Athletics	812
Sporting Goods	Outdoor Recreation	5642

At the bottom of the screen, there is a status bar with the text "For Help, press F1", the date and time "11/4/2015 12:59:21 PM\*", and the dimensions "19 X 3".

# 41. QlikView – Pivot Tables

Pivot Tables are widely used in data analysis to present sum of values across many dimensions available in the data. QlikView's Chart option has the feature to create a Pivot Table by choosing the appropriate chart type.

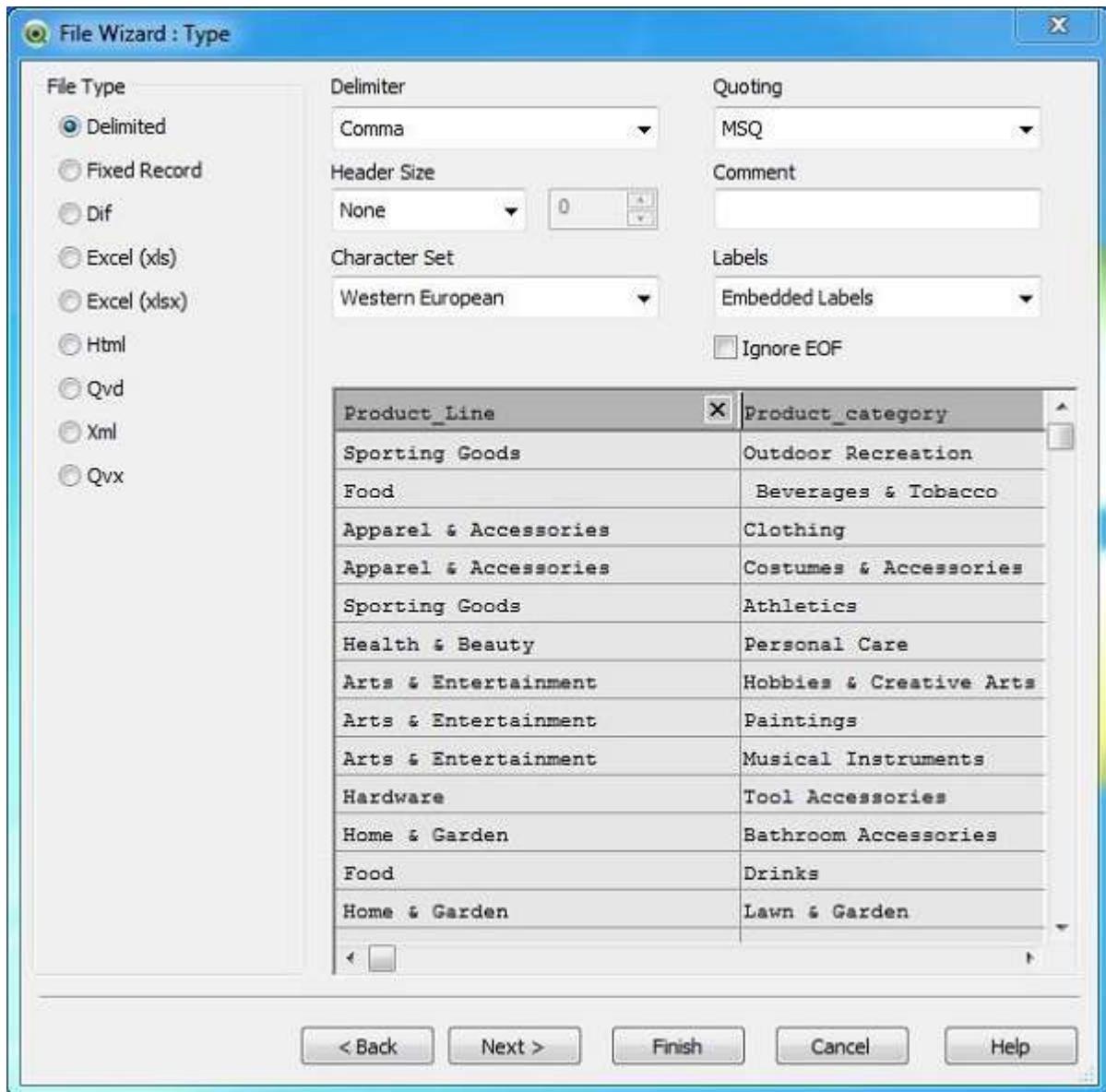
## Input Data

Let us consider the following input data, which represents the sales figure of different product lines and product categories.

Product_Line	Product_category	Value
Sporting Goods	Outdoor Recreation	5642
Food	Beverages & Tobacco	2514
Apparel & Accessories	Clothing	2365
Apparel & Accessories	Costumes & Accessories	4487
Sporting Goods	Athletics	812
Health & Beauty	Personal Care	6912
Arts & Entertainment	Hobbies & Creative Arts	5201
Arts & Entertainment	Paintings	8451
Arts & Entertainment	Musical Instruments	1245
Hardware	Tool Accessories	456
Home & Garden	Bathroom Accessories	241
Food	Drinks	1247
Home & Garden	Lawn & Garden	5462
Office Supplies	Presentation Supplies	577
Hardware	Blocks	548
Baby & Toddler	Diapering	1247
Baby & Toddler	Toys	257
Home & Garden	Pipes	1241
Office Supplies	Dispaly Borad	2177

## Load Script

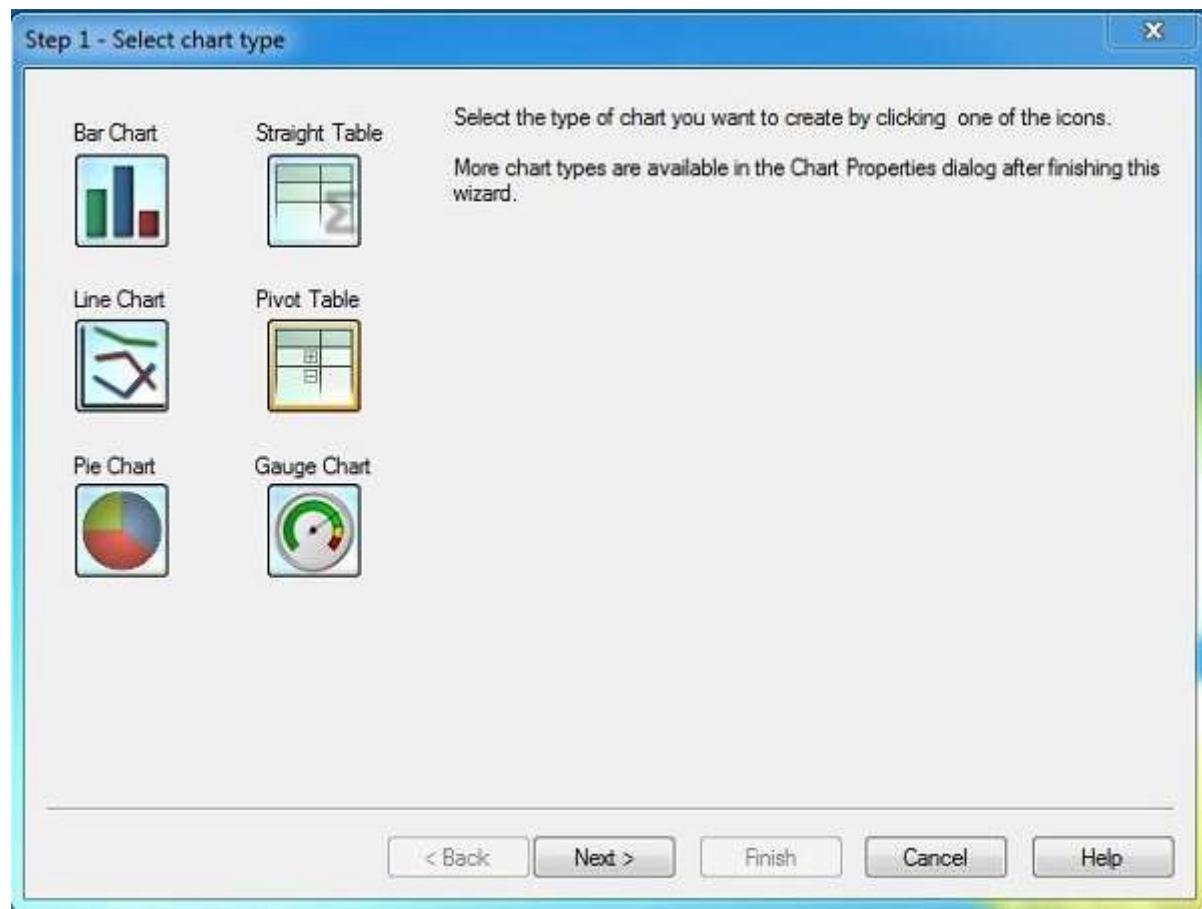
The above data is loaded to the QlikView's memory by using the script editor. Open the Script editor from the File menu or press **Control+E**. Choose the "Table Files" option from the "Data from Files" tab and browse for the file containing the above data. The following screen appears.



Click "OK" and press "Control+R" to load the data into QlikView's memory.

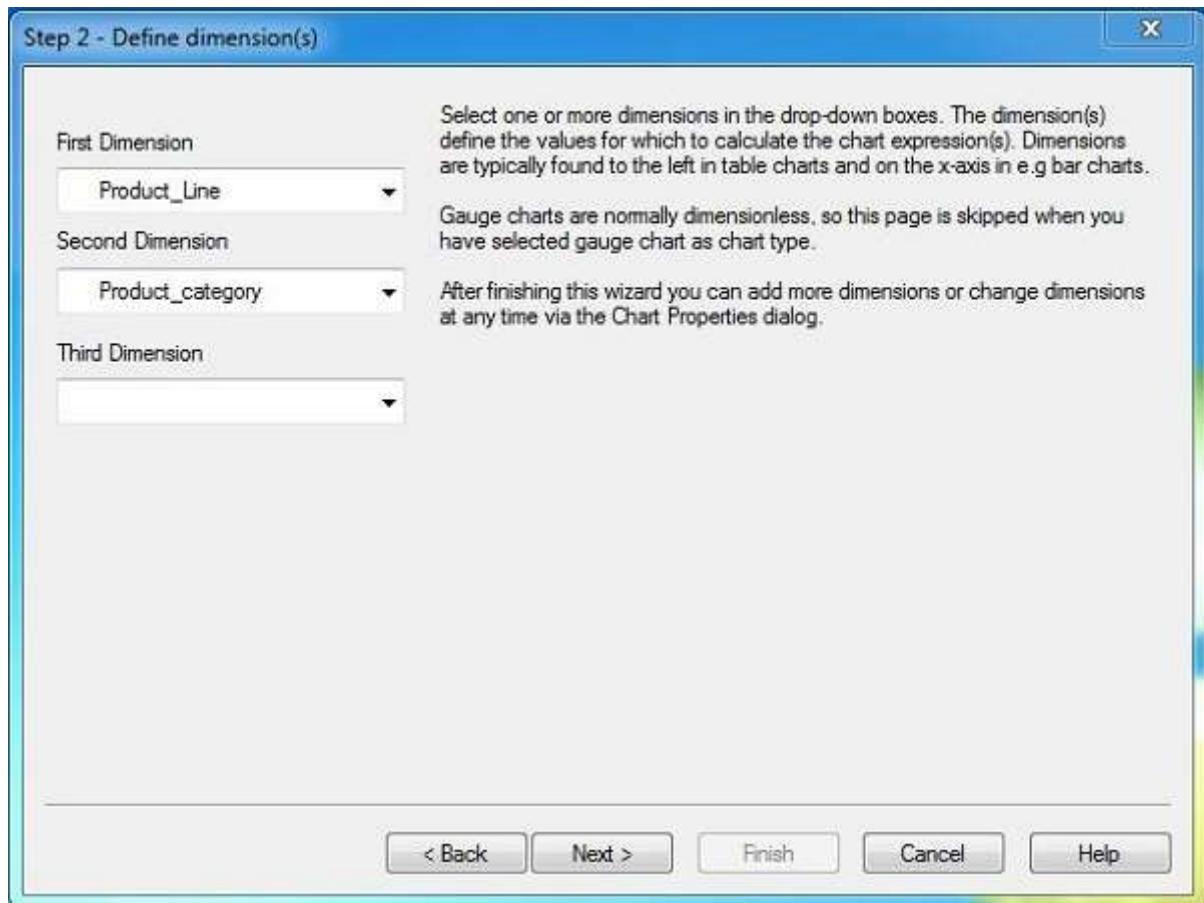
## Select Chart Type

Next, we use the chart wizard to select the **Pivot Table** option. Click Next.



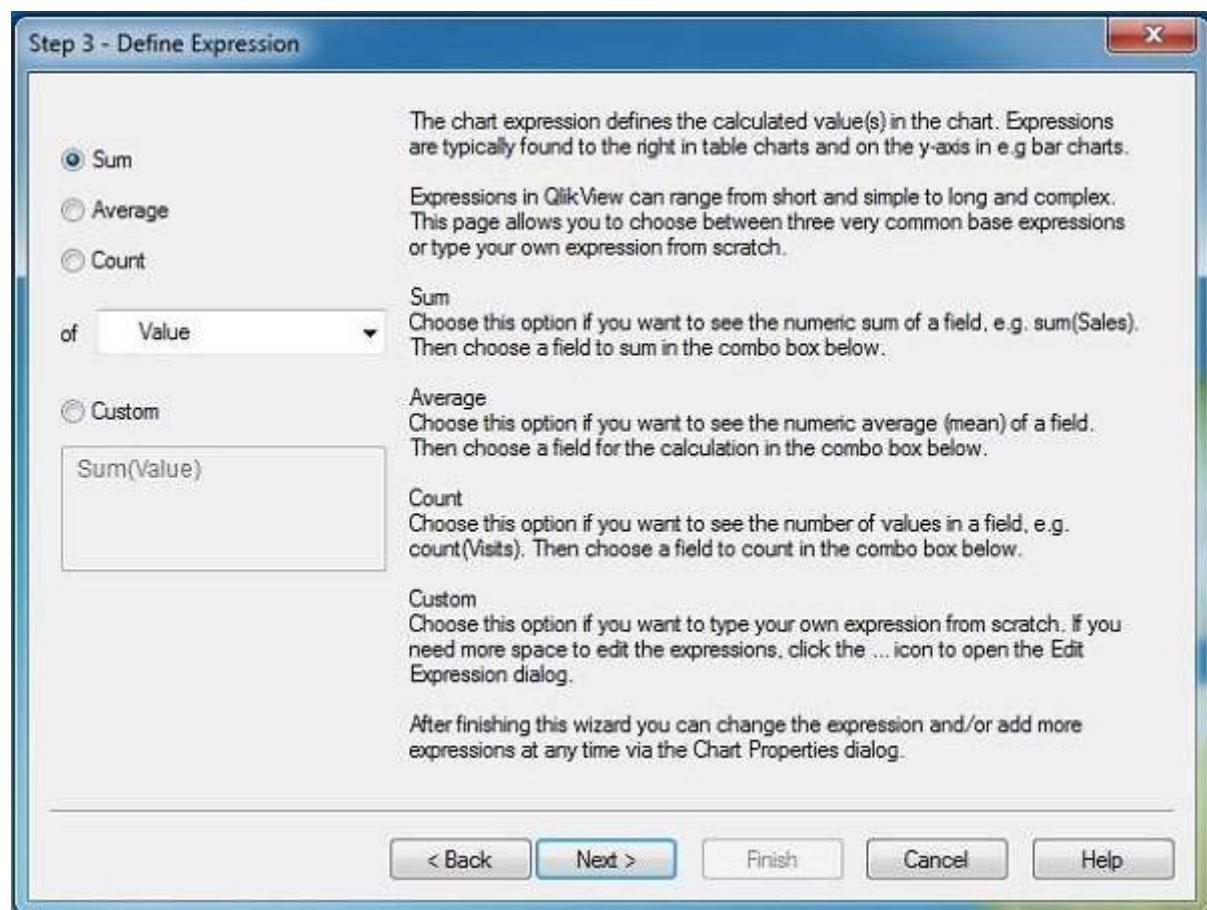
## Select Chart Dimension

In the next screen, we choose Product\_Line as the first dimension for the chart.



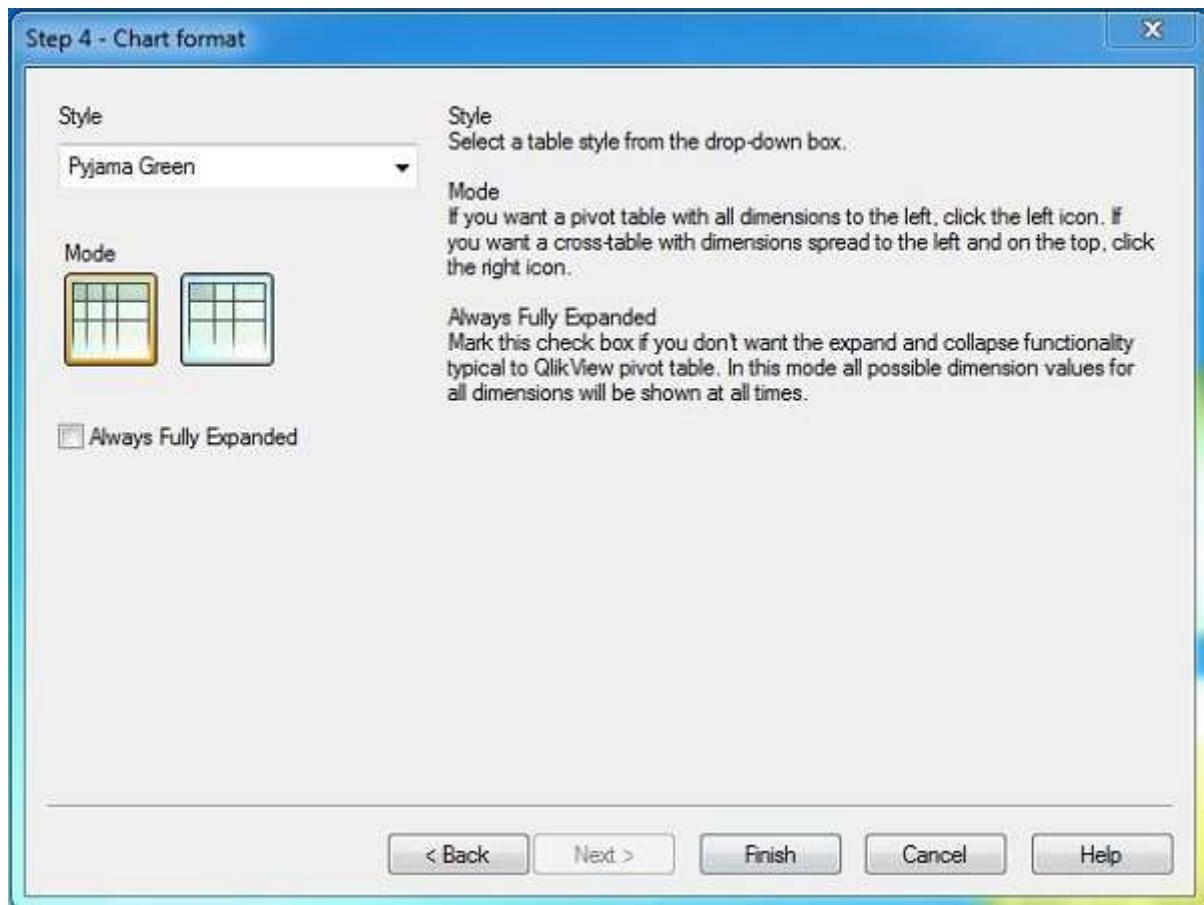
## Select Chart Expression

The next screen prompts us for selecting the chart expression where we choose the sum of value.



## Select the Chart Format

On clicking next, we get the screen to choose chart format in which we select **Pyjama Green** as the style and the default mode.



## Pivot Chart Data

Completing the above steps gives us the final chart as below.

The screenshot shows the QlikView x64 Personal Edition interface with a window titled "QlikView x64 Personal Edition - [C:\Qlikview\files\pivot\_table.qvw\*]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. The toolbar contains various icons for file operations, search, and navigation. The main area is labeled "Main" and displays a pivot table with the following data:

Product_Line	Product_category	Sum(Value)
Sporting Goods		6454
Food		3761
Apparel & Accessories	Clothing	2365
	Costumes & Accessories	4487
Health & Beauty		6912
Arts & Entertainment		14897
Hardware		1004
	Bathroom Accessories	241
Home & Garden	Lawn & Garden	5462
	Pipes	1241
Office Supplies		2754
Baby & Toddler		1504

At the bottom left, it says "For Help, press F1". At the bottom right, it shows the date and time "11/4/2015 1:23:02 PM\*" and the window size "12 X 3".

## 42. QlikView – Set Analysis

QlikView's Set Analysis feature is used to segregate the data in different sheet objects into many sets and keeps the values unchanged in some of them. In simpler terms, it creates an option to not associate some sheet objects with others while the default behavior is all sheet objects get associated with each other. This helps in filtering the data in one sheet object and seeing the corresponding result in others, while the sheet object chosen as a different set displays values as per its own filters.

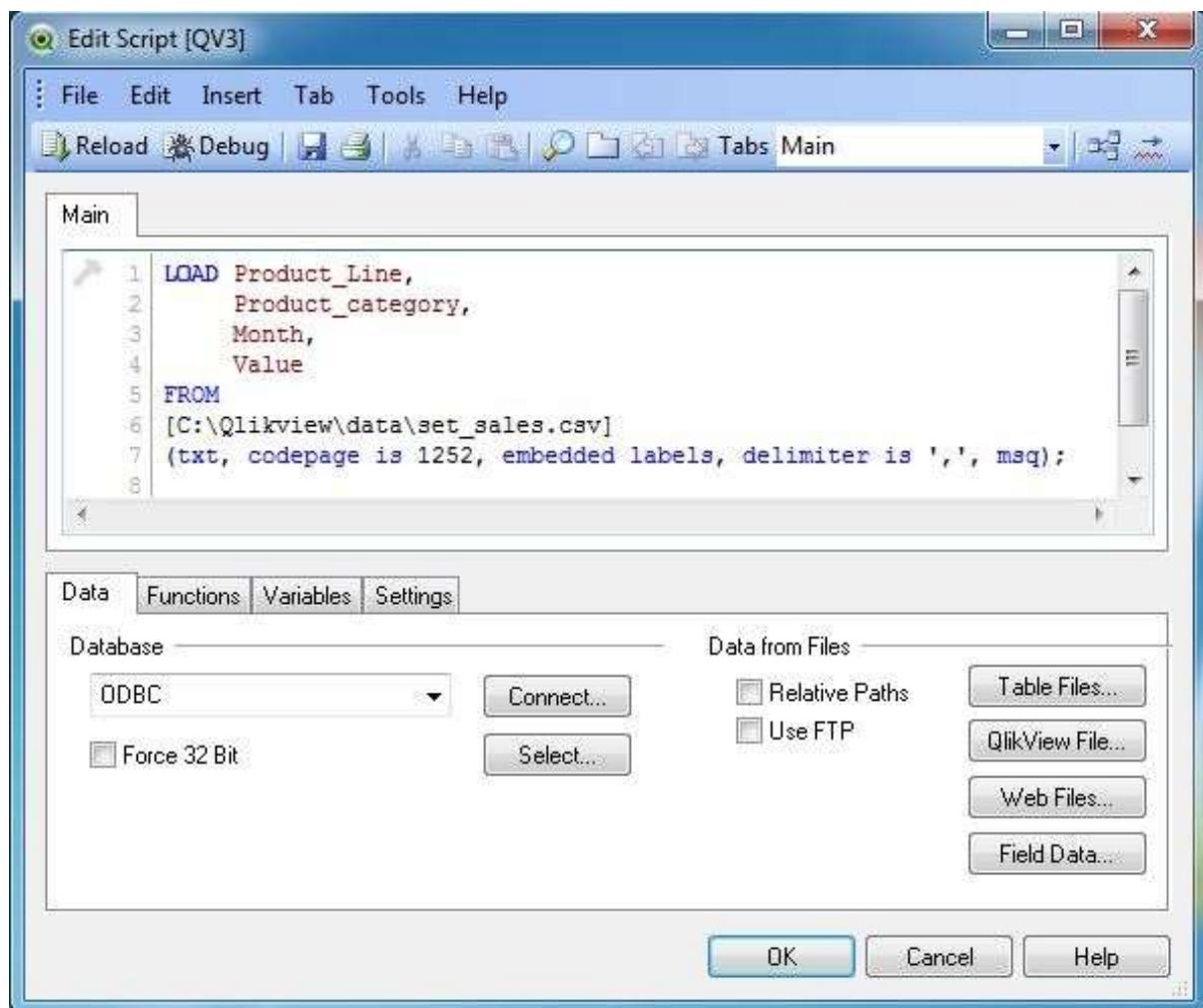
### Input Data

Let us consider the following input data, which represents the sales figure of different product lines and product categories.

```
Product_Line,Product_category,Month,Value
Arts & Entertainment,Hobbies & Creative Arts,Jan,5201
Arts & Entertainment,Paintings,Feb,8451
Arts & Entertainment,Musical Instruments,Jan,1245
Baby & Toddler,Diapering,Mar,1247
Baby & Toddler,Toys,Dec,257
Apparel & Accessories,Clothing,Feb,574
Apparel & Accessories,Costumes & Accessories,Apr,1204
Arts & Entertainment,Musical Instruments,Apr,3625
Baby & Toddler,Diapering,Apr,1281
Apparel & Accessories,Clothing,Jul,2594
Arts & Entertainment,Paintings,Sep,6531
Baby & Toddler,Toys,May,7421
Apparel & Accessories,Clothing,Aug,2541
Arts & Entertainment,Paintings,Oct,2658
Arts & Entertainment,Musical Instruments,Mar,1185
Baby & Toddler,Diapering,Jun,1209
```

## Load Script

The above data is loaded to QlikView memory by using the script editor. Open the Script editor from the File menu or press **Control+E**. Choose the "Table Files" option from the "Data from Files" tab and browse for the file containing the above data. A screen appears as shown below.



## Create Table Box

Choose all the fields available to create a table box using the menu option **Layout -> New Sheet Objects -> Table Box** and a list box containing the month's field using the menu option **Layout -> New Sheet Objects -> List Box**. Also, create a straight table chart showing the total sales under each product category.

The screenshot shows the QlikView interface with the following components:

- Main Sheet:** Contains two table boxes and one list box.
  - Table Box 1:** Shows sales data with columns: Product\_Line, Product\_category, Month, and Value. The data includes rows for Apparel & Accessories, Arts & Entertainment, Baby & Toddler, and others across various months.
  - Table Box 2:** Shows Sales Sum with columns: Product\_category and Sum(Value). The data includes rows for Hobbies & Creative Arts, Paintings, Musical Instruments, Diapering, Toys, Clothing, and Costumes & Accessories.
  - List Box:** Shows a list of months: Apr, Aug, Dec, Feb, Jan, Jul, Jun, Mar, May, Oct, and Sep.
- Status Bar:** Displays "For Help, press F1" and the date/time "11/9/2015 7:35:04 AM".

## Data Association

Now we can observe the association between these three sheet objects by selecting some values in one of them. Let us select the month Apr and Jan from the Month list Box. We can see the change in values in the Table Box and chart showing the related values.

The screenshot shows the QlikView interface with a 'Main' sheet active. At the top, there is a menu bar with File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. Below the menu is a toolbar with various icons. The main area contains two objects: a list box titled 'Month' and a table titled 'Sales Sum'.

**Month List Box:**

Month

**Sales Sum Table:**

Product_category	Sum(Value)
Hobbies & Creative Arts	5201
Musical Instruments	4870
Diapering	1281
Costumes & Accessories	1204
<b>12556</b>	

At the bottom of the screen, there is a status bar with the text "For Help, press F1", the date and time "11/9/2015 7:35:04 AM\*", and page information "D: 2/11 F: 5/16".

## Clone Object

Next, we clone the sales sum chart to produce a new set of data not associated with other sheet objects. Right click on the chart **Sales Sum** and click on the option **Clone** as shown below. Another copy of the same chart appears in the QlikView document.

The screenshot shows the QlikView interface with the title bar "QlikView x64 Personal Edition - [C:\Users\vyaapak3\Downloads\set\_analysis.qvw\*]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. Below the menu is a toolbar with various icons. The main area is titled "Main" and contains two charts. The left chart is titled "Sales Sum" and has columns: Product\_category, Sum, and Value. The right chart has columns: Month, Value, and Sum(Value). A context menu is open over the "Sales Sum" chart, with the "Clone" option highlighted in yellow. Other options in the menu include Properties..., Detach, Fit Columns to Data, Equal Column Width, Sort, Clear All Selections, Print..., Send to Excel, Export..., Copy to Clipboard, Minimize, Maximize, Help, and Remove.

Product_line	Product_category	Month	Value
Apparel & Accessories	Clothing	Aug	2541
Apparel & Accessories	Clothing	Feb	574
Apparel & Accessories	Clothing	Jul	2594
Apparel & Accessories	Costumes & Accessories	Apr	1204
Arts & Entertainment	Hobbies	Properties...	
Arts & Entertainment	Musical Instruments		
Arts & Entertainment	Musical Instruments		
Arts & Entertainment	Musical Instruments		
Arts & Entertainment	Paintings		
Arts & Entertainment	Paintings		
Baby & Toddler	Diapering		
Baby & Toddler	Diapering		
Baby & Toddler	Diapering		
Baby & Toddler	Toys		
Baby & Toddler	Toys		

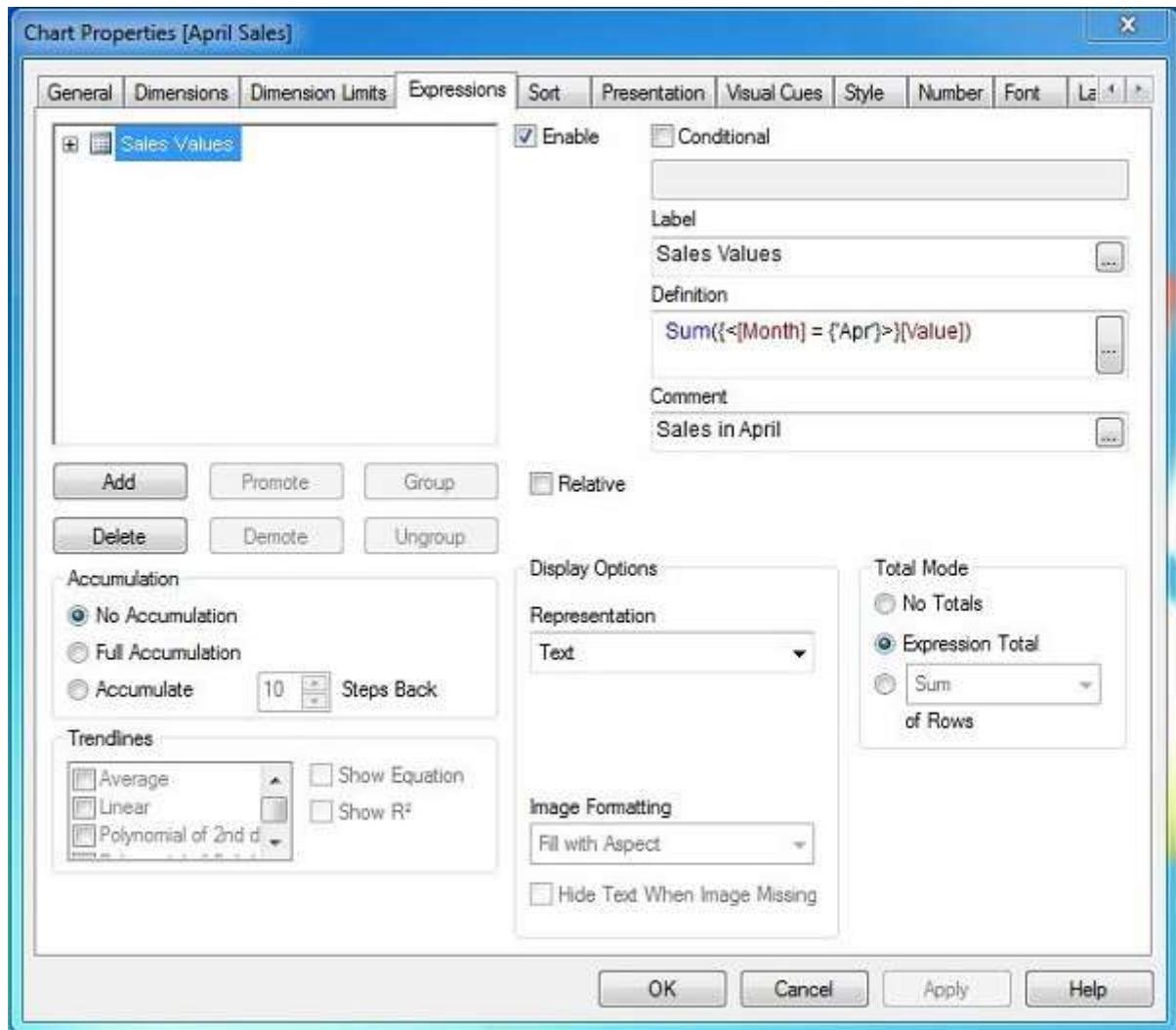
Month	Value
Apr	
Aug	
Dec	
Feb	
Jan	
Jul	
Jun	
Mar	
May	
Oct	
Sep	

Product_category	Sum
Hobbies & Creative Arts	47224
Paintings	5201
Musical Instruments	17640
Diapering	6055
Toys	3737
Clothing	7678
Costumes & Accessories	5709
	1204

Create a copy of this chart      11/9/2015 7:35:04 AM\*      7 X 2

## Set Expression

Next, we choose the second copy of the chart **Sales Sum** and right click it to get the chart properties. We create an expression called Sales values writing the formula under the **Definition** tab as shown below.



## Applying Set Analysis

On completing the above given steps, we find that when we select the month June we get the associated values in the Table Box and Sales Sum chart. However, the April sales does not change as it is based on the data from the set expression.

The screenshot shows the QlikView interface with three main components:

- Table Box:** Displays a list of months with their corresponding values. The month "Jun" is highlighted in green, indicating it is selected. The data is as follows:

Month	Value
Apr	
Aug	
Dec	
Feb	
Jan	
Jul	
<b>Jun</b>	<b>1209</b>
Mar	
May	
Oct	
Sep	

- Table Box:** Shows the total sales sum for the selected product category. The data is as follows:

Sales Sum	Product_category	Sum(Value)
	Diapering	<b>1209</b>

- Table Box:** Shows the sales values for April across different product categories. The data is as follows:

April Sales	Product_category	Sales Values
	Musical Instruments	3625
	Diapering	1281
	Costumes & Accessories	1204
<b>6110</b>		

## 43. QlikView – Joins

Joins in QlikView are used to combine data from two data sets into one. Joins in QlikView mean the same as in joins in SQL. Only the column and row values that match the join conditions are shown in the output. In case you are completely new to joins, you may like to first learn about them [here](#).

### Input Data

Let us consider the following two CSV data files, which are used as input for further illustrations.

Product List:

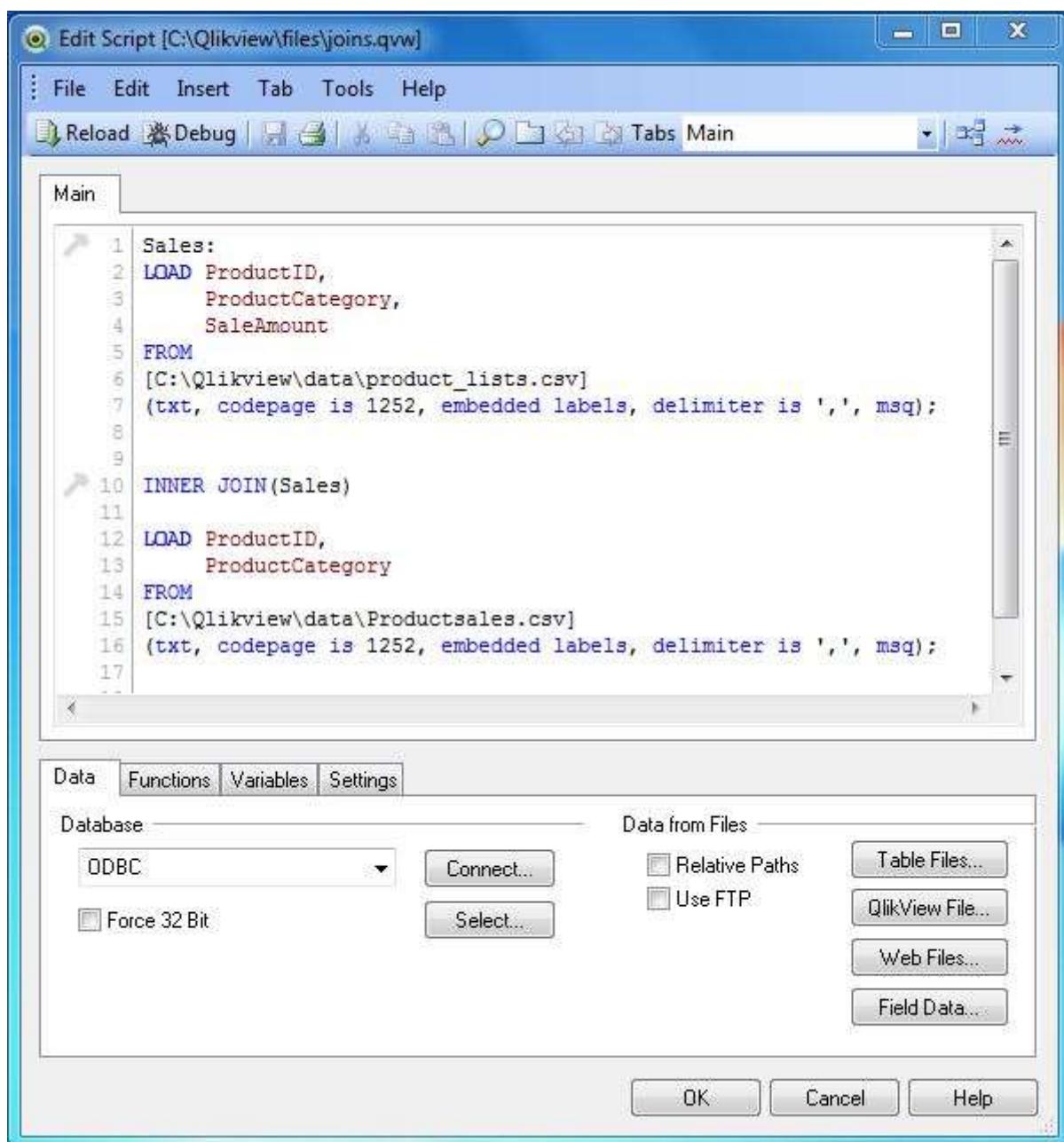
```
ProductID,ProductCategory  
1,Outdoor Recreation  
2,Clothing  
3,Costumes & Accessories  
4,Athletics  
5,Personal Care  
6,Hobbies & Creative Arts
```

ProductSales:

```
ProductID,ProductCategory,SaleAmount  
4,Athletics,1212  
5,Personal Care,5211  
6,Hobbies & Creative Arts,1021  
7,Dispaly Borad,2177  
8,Game,1145  
9,soap,1012  
10,Beverages & Tobacco,2514
```

## Inner Join

We load the above input data using the script editor, which is invoked by pressing **Control+E**. Choose the option **Table Files** and browse for the Input file. Then we edit the commands in the script to create an inner join between the tables.



The screenshot shows the QlikView Script Editor window titled "Edit Script [C:\Qlikview\files\joins.qvw]". The main pane displays the following QlikView script:

```

1 Sales:
2   LOAD ProductID,
3     ProductCategory,
4     SaleAmount
5   FROM
6     [C:\Qlikview\data\product_lists.csv]
7     (txt, codepage is 1252, embedded labels, delimiter is ',', msq);
8
9
10 INNER JOIN(Sales)
11
12   LOAD ProductID,
13     ProductCategory
14   FROM
15     [C:\Qlikview\data\Productsales.csv]
16     (txt, codepage is 1252, embedded labels, delimiter is ',', msq);
17

```

The "Data" tab is selected in the bottom navigation bar. The "Database" section contains an "ODBC" dropdown menu, a "Connect..." button, and a "Force 32 Bit" checkbox. The "Data from Files" section contains buttons for "Table Files...", "QlikView File...", "Web Files...", and "Field Data...". At the bottom are "OK", "Cancel", and "Help" buttons.

Inner join fetches only those rows, which are present in both the tables. In this case, the rows available in **both Product List and Product Sales** table are fetched. We create a **Table Box** using the menu **Layout -> New Sheet Objects -> Table Box** where we choose all the three fields - ProductID, ProductCategory and SaleAmount to be displayed.

INNER JOIN		
ProductID	ProductCategory	SaleAmount
4	Athletics	1212
5	Personal Care	5211
6	Hobbies & Creative Arts	1021

## Left Join

Left join involves fetching all the rows from the table in the left and the matching rows from the table in the right.

### Load Script

```

Sales:
LOAD ProductID,
    ProductCategory,
    SaleAmount
FROM
[C:\Qlikview\data\product_lists.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);

LEFT JOIN(Sales)

LOAD ProductID,
    ProductCategory
FROM
[C:\Qlikview\data\Productsales.csv]

```

```
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);
```

We create a **Table Box** using the menu **Layout -> New Sheet Objects -> Table Box**, where we choose all the three fields - ProductID, ProductCategory and SaleAmount to be displayed.

The screenshot shows the QlikView interface with a window titled "QlikView x64 Personal Edition - [C:\Qlikview\files\joins.qvw]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. The toolbar below has various icons for file operations like Open, Save, Print, and zoom. The main area is labeled "Main" and contains a table box titled "LEFT JOIN". The table has three columns: ProductID, ProductCategory, and SaleAmount. The data is as follows:

ProductID	ProductCategory	SaleAmount
4	Athletics	1212
5	Personal Care	5211
6	Hobbies & Creative Arts	1021
7	Display Board	2177
8	Game	1145
9	soap	1012
10	Beverages & Tobacco	2514

At the bottom of the interface, there is a status bar with "For Help, press F1", the date and time "11/21/2015 7:18:07 PM", and a "7 X 3" grid icon.

## Right Join

Right join involves fetching all the rows from the table in the right and the matching rows from the table in the left.

### Load Script

```
Sales:
LOAD ProductID,
    ProductCategory,
    SaleAmount
FROM
[C:\Qlikview\data\product_lists.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);
RIGHT JOIN(Sales)

LOAD ProductID,
    ProductCategory
```

```
FROM
[C:\Qlikview\data\Productsales.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);
```

We create a **Table Box** using the menu **Layout -> New Sheet Objects -> Table Box**, where we choose all the three fields - ProductID, ProductCategory and SaleAmount to be displayed.

RIGHT JOIN		
ProductID	ProductCategory	SaleAmount
1	Outdoor Recreation	-
2	Clothing	-
3	Costumes & Accessories	-
4	Athletics	1212
5	Personal Care	5211
6	Hobbies & Creative Arts	1021

## Outer Join

Outer join involves fetching all the rows from the table in the right as well as from the table in the left.

### Load Script

```
Sales:
LOAD ProductID,
    ProductCategory,
    SaleAmount
FROM
[C:\Qlikview\data\product_lists.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);
OUTER JOIN(Sales)
LOAD ProductID,
```

```

ProductCategory
FROM
[C:\Qlikview\data\Productsales.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);

```

We create a **Table Box** using the menu **Layout -> New Sheet Objects -> Table Box** where we choose all the three fields - ProductID, ProductCategory and SaleAmount to be displayed.

The screenshot shows the QlikView x64 Personal Edition interface. The title bar reads "QlikView x64 Personal Edition - [C:\Qlikview\files\joins.qvw]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. The toolbar below the menu bar contains various icons for file operations like Open, Save, Print, and zoom. The main workspace is titled "Main" and contains a table titled "OUTER JOIN". The table has three columns: "ProductID", "ProductCategory", and "SaleAmount". The data is as follows:

ProductID	ProductCategory	SaleAmount
1	Outdoor Recreation	-
2	Clothing	-
3	Costumes & Accessories	-
4	Athletics	1212
5	Personal Care	5211
6	Hobbies & Creative Arts	1021
7	Dispaly Borad	2177
8	Game	1145
9	soap	1012
10	Beverages & Tobacco	2514

At the bottom of the interface, there is a status bar with the text "For Help, press F1", the date and time "11/21/2015 7:20:35 PM", and the dimensions "10 X 3".

## 44. QlikView – Keeps

The keep command in QlikView is used to combine data from two data sets keeping both the data sets available in memory. It is very similar to joins we covered in the previous chapter except for two major differences. First difference is - in case of keep; both the datasets are available in QlikView's memory while in join the load statements produce only one data set from which you have to choose the columns. The second difference being - there is no concept of outer keep where as we have outer join available in case of joins.

### **Input Data**

Let us consider the following two CSV data files, which are used as input for further illustrations.

Product List:

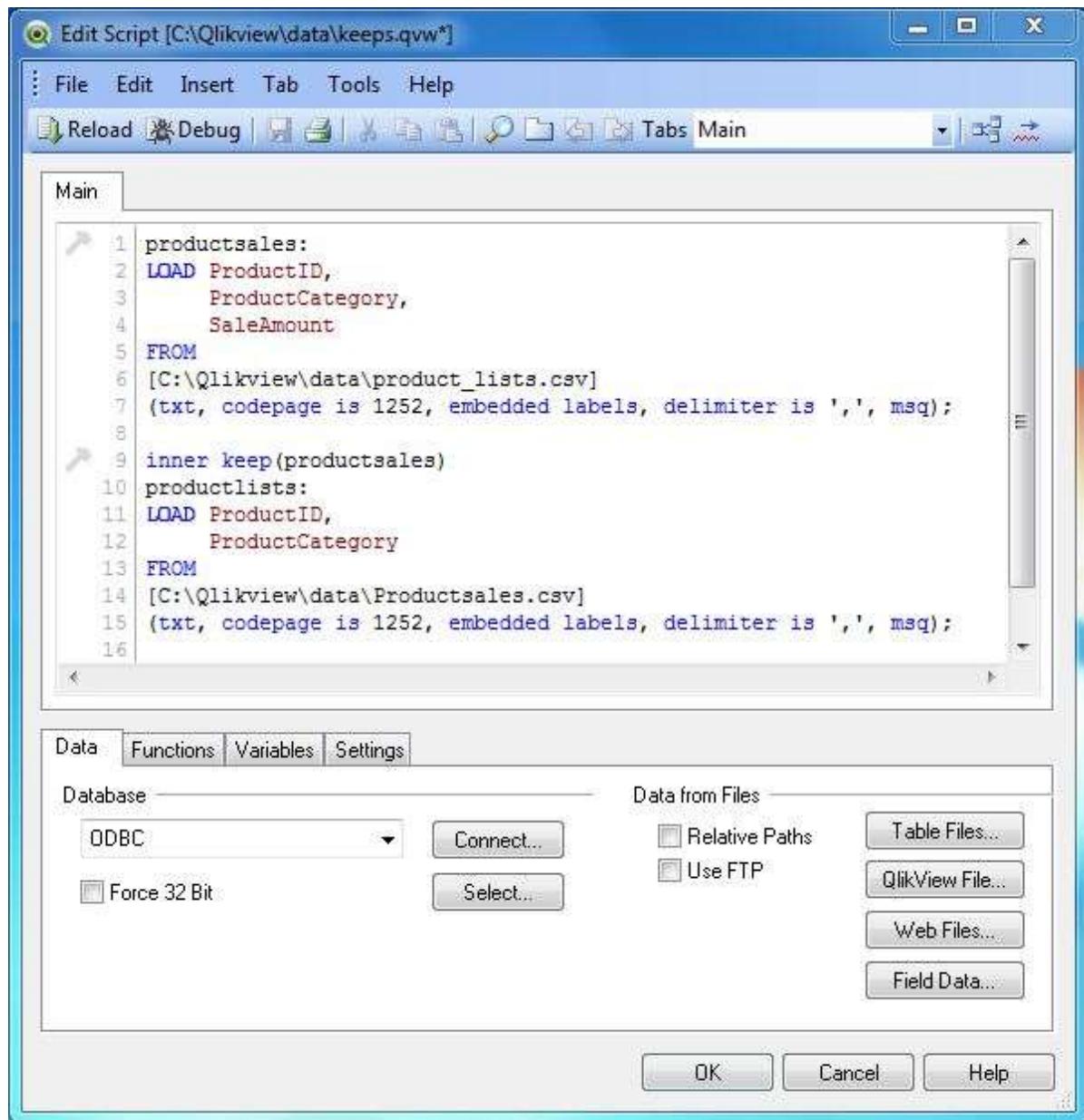
```
ProductID,ProductCategory  
1,Outdoor Recreation  
2,Clothing  
3,Costumes & Accessories  
4,Athletics  
5,Personal Care  
6,Hobbies & Creative Arts
```

Product Sales:

```
ProductID,ProductCategory,SaleAmount  
4,Athletics,1212  
5,Personal Care,5211  
6,Hobbies & Creative Arts,1021  
7,Dispaly Borad,2177  
8,Game,1145  
9,soap,1012  
10,Beverages & Tobacco,2514
```

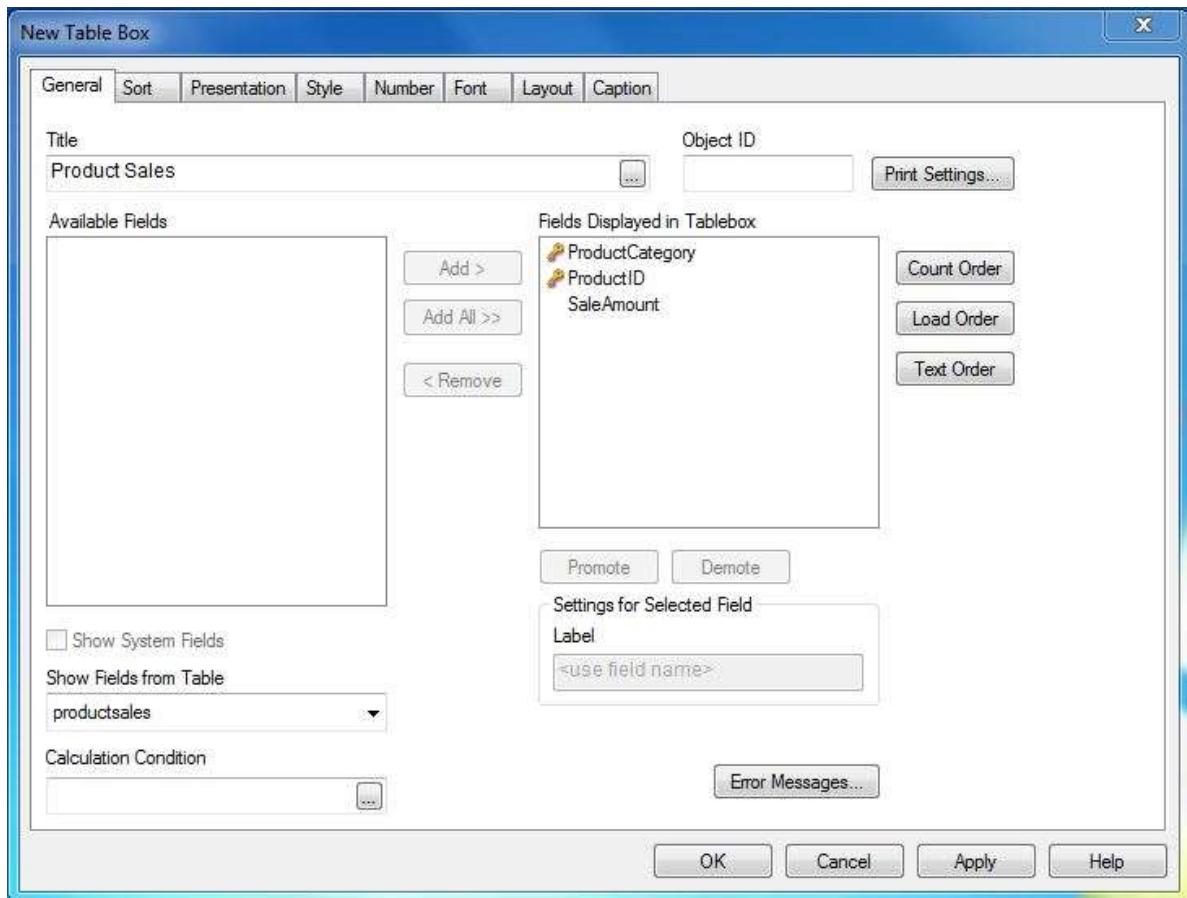
## Inner Keep

We load the above input data using the script editor, which is invoked by pressing **Control+E**. Choose the option **Table Files** and browse for the Input file. Then we edit the commands in the script to create an inner keep between the tables.

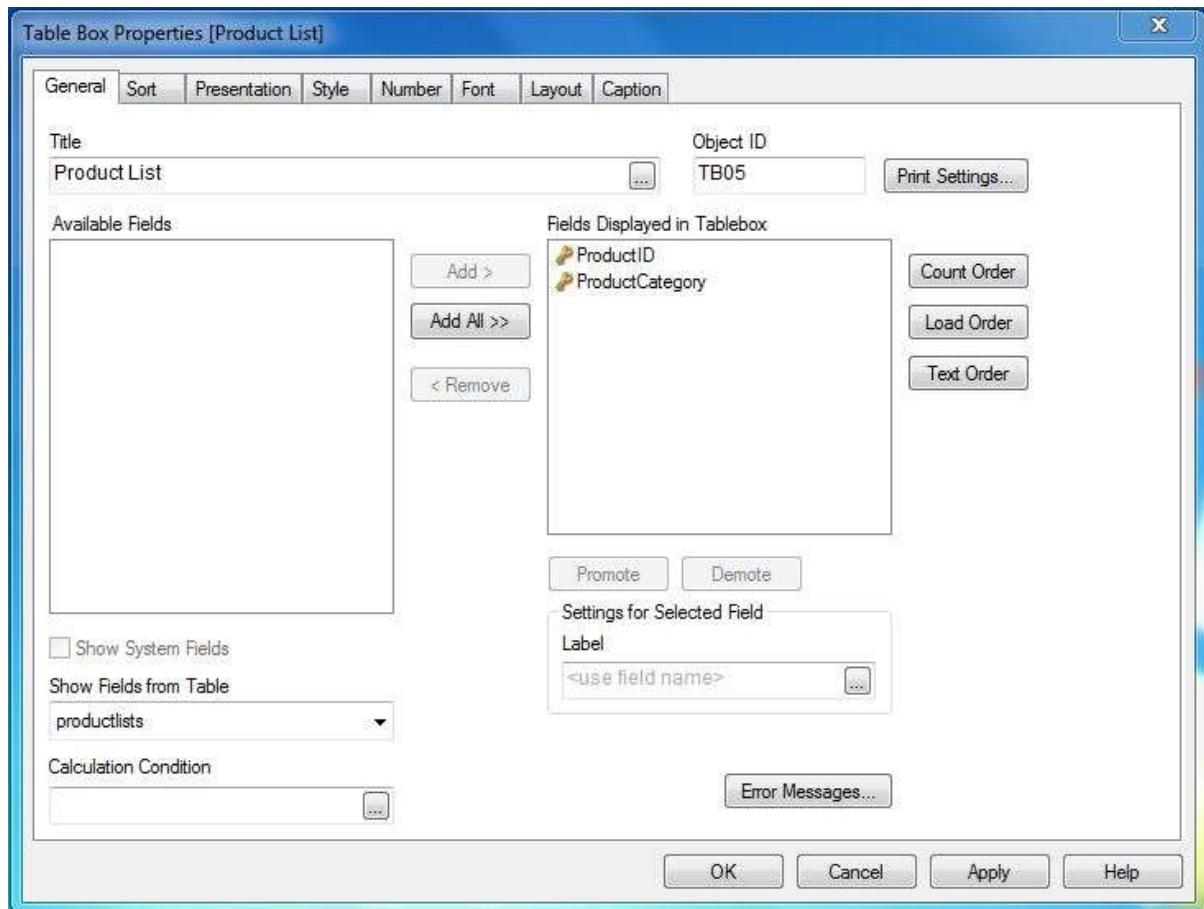


Inner keep fetches only those rows, which are present in both the tables. In this case, the rows available in **both Product List and Product Sales** table are fetched. We create a Table Boxes using the menu **Layout -> New Sheet Objects -> Table Box**.

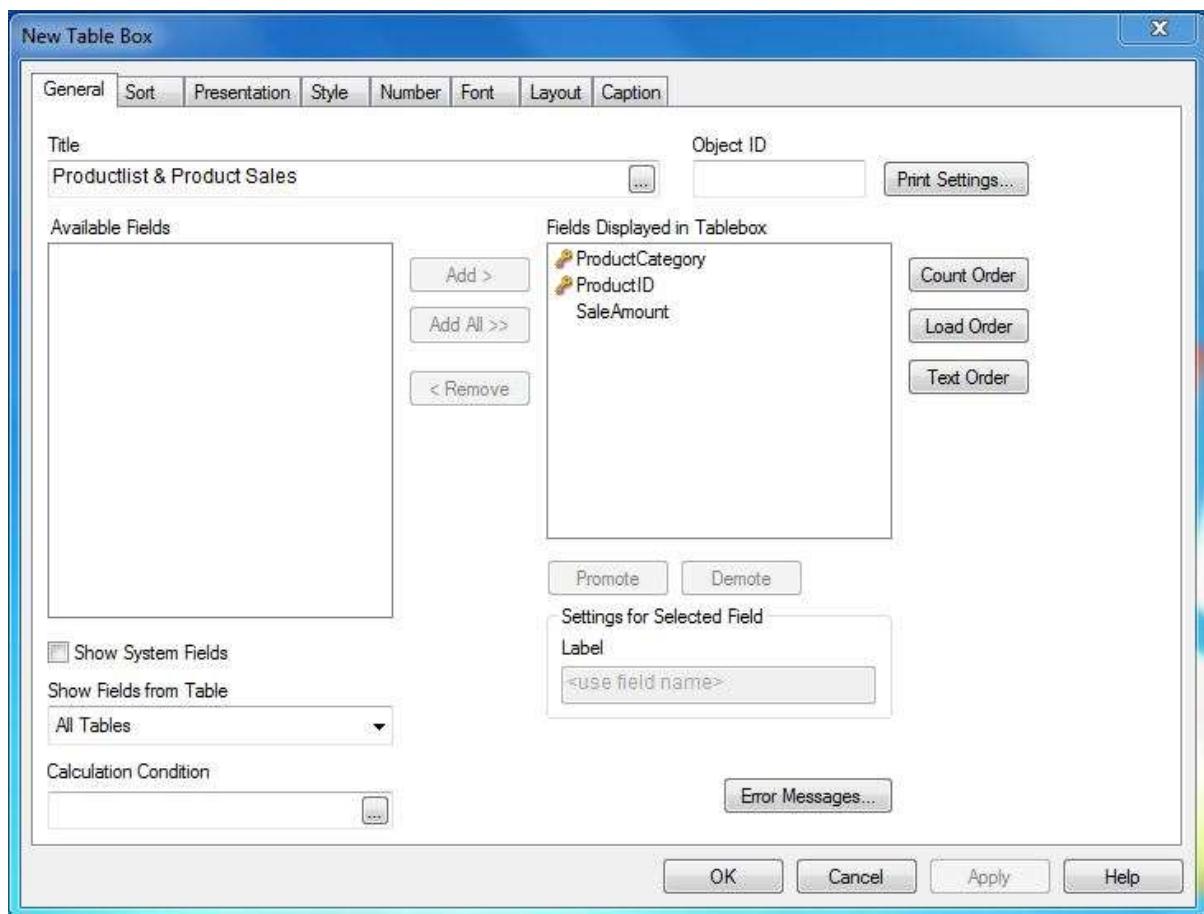
First, we choose only the **productSales** table, which gives us the fields - ProductID, ProductCategory and SaleAmount to be displayed.



Next, we choose the **ProductList** data set, which gives us the fields ProductID and ProductCategory.



Finally, we choose the **All Tables** option and get all the available fields from all the tables.



The following report shows all the Tables Boxes from the above given steps.

The screenshot shows the QlikView interface with three tables displayed:

- Product Sales**: Shows sales data with columns ProductID, ProductCategory, and SaleAmount. The data is:

ProductID	ProductCategory	SaleAmount
4	Athletics	1212
5	Personal Care	5211
6	Hobbies & Creative Arts	1021

- Product List**: Shows a list of categories with columns ProductID and ProductCategory. The data is:

ProductID	ProductCategory
4	Athletics
5	Personal Care
6	Hobbies & Creative Arts

- Productlist & Product Sa...**: A joined table with columns Prod... (ProductID), ProductCategory, and SaleAmount. The data is identical to the Product Sales table.

At the bottom left, it says "For Help, press F1". At the bottom right, it shows the date and time: "11/22/2015 7:19:26 AM".

## Left Keep

Left keep is similar to left join, which keeps all the rows from the table in the left along with both the data set being available in QlikView's memory.

### Left keep Script

The following script is used to create the resulting data sets with left keep command.

```
productsales:
LOAD ProductID,
    ProductCategory,
    SaleAmount
FROM
[C:\Qlikview\data\product_lists.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);
left keep(productsales)
```

```
productlists:
LOAD ProductID,
    ProductCategory
FROM
[C:\Qlikview\data\Productsales.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);
```

## Left keep Data

When we change the script as above and refresh the data in the report using **Control+R**, we get the following data in the sheet objects.

The screenshot shows the QlikView interface with three sheet objects:

- Product Sales**: A table with columns ProductID, ProductCategory, and SaleAmount. The data is:

ProductID	ProductCategory	SaleAmount
4	Athletics	1212
5	Personal Care	5211
6	Hobbies & Creative Arts	1021
7	Display Board	2177
8	Game	1145
9	soap	1012
10	Beverages & Tobacco	2514

- Product List**: A table with columns ProductID and ProductCategory. The data is:

ProductID	ProductCategory
4	Athletics
5	Personal Care
6	Hobbies & Creative Arts
7	Display Board
8	Game
9	soap
10	Beverages & Tobacco

- Productlist & Product Sales**: A table with columns Prod..., ProductCategory, and SaleAmount. The data is identical to the Product Sales table.

At the bottom left, it says "For Help, press F1". At the bottom right, it shows the date and time: "11/22/2015 7:14:31 AM".

## Right Keep

Right keep is similar to left join, which keeps all the rows from the table in the right along with both the data set being available in QlikView's memory.

## Right keep Script

The following script is used to create the resulting data sets with left keep command.

```

productsales:
LOAD ProductID,
    ProductCategory,
    SaleAmount
FROM
[C:\Qlikview\data\product_lists.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);

right keep(productsales)
productlists:
LOAD ProductID,
    ProductCategory
FROM
[C:\Qlikview\data\Productsales.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);

```

## Right Keep Data

When we change the script as above and refresh the data in the report using **Control+R**, we get the following data in the sheet objects.

QlikView x64 Personal Edition - [C:\Qlikview\data\keeps.qvw]

Main

**Product Sales**

ProductID	ProductCategory	SaleAmount
1	Outdoor Recreation	-
2	Clothing	-
3	Costumes & Accessor...	-
4	Athletics	1212
5	Personal Care	5211
6	Hobbies & Creative Arts	1021

**Product List**

ProductID	ProductCategory
1	Outdoor Recreation
2	Clothing
3	Costumes & Accessories
4	Athletics
5	Personal Care
6	Hobbies & Creative Arts

**Productlist & Product Sales**

Prod...	ProductCategory	SaleAmount
1	Outdoor Recreation	-
2	Clothing	-
3	Costumes & Accessories	-
4	Athletics	1212
5	Personal Care	5211
6	Hobbies & Creative Arts	1021

For Help, press F1      11/22/2015 7:17:28 AM

## 45. QlikView – Concatenation

Concatenation feature in QlikView is used to append the rows from one table to another. It happens even when the tables have different number of columns. It differs from both Join and Keep command, as it does not merge the matching rows from two tables into one row.

### **Input Data**

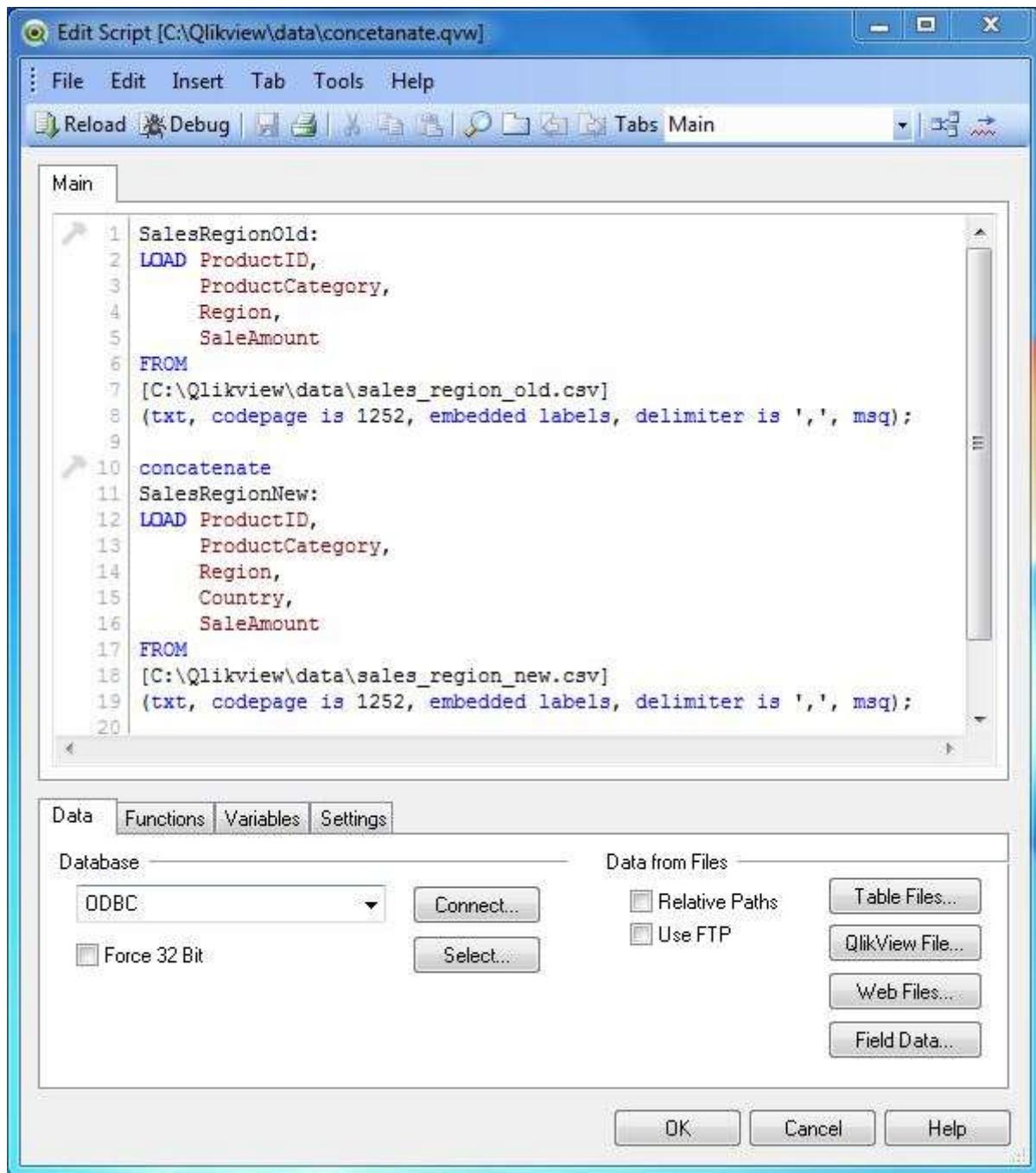
Let us consider the following two CSV data files, which are used as input for further illustrations. Please note the second data set has an additional column named Country.

```
SalesRegionOld.csv  
ProductID,ProductCategory,Region,SaleAmount  
1,Outdoor Recreation,Europe,4579  
2,Clothing,Europe,4125  
3,Costumes & Accessories,South Asia,6521  
4,Athletics,South Asia,4125  
5,Personal Care,Australia,5124  
6,Arts & Entertainment,North America,1245  
7,Hardware,South America,456
```

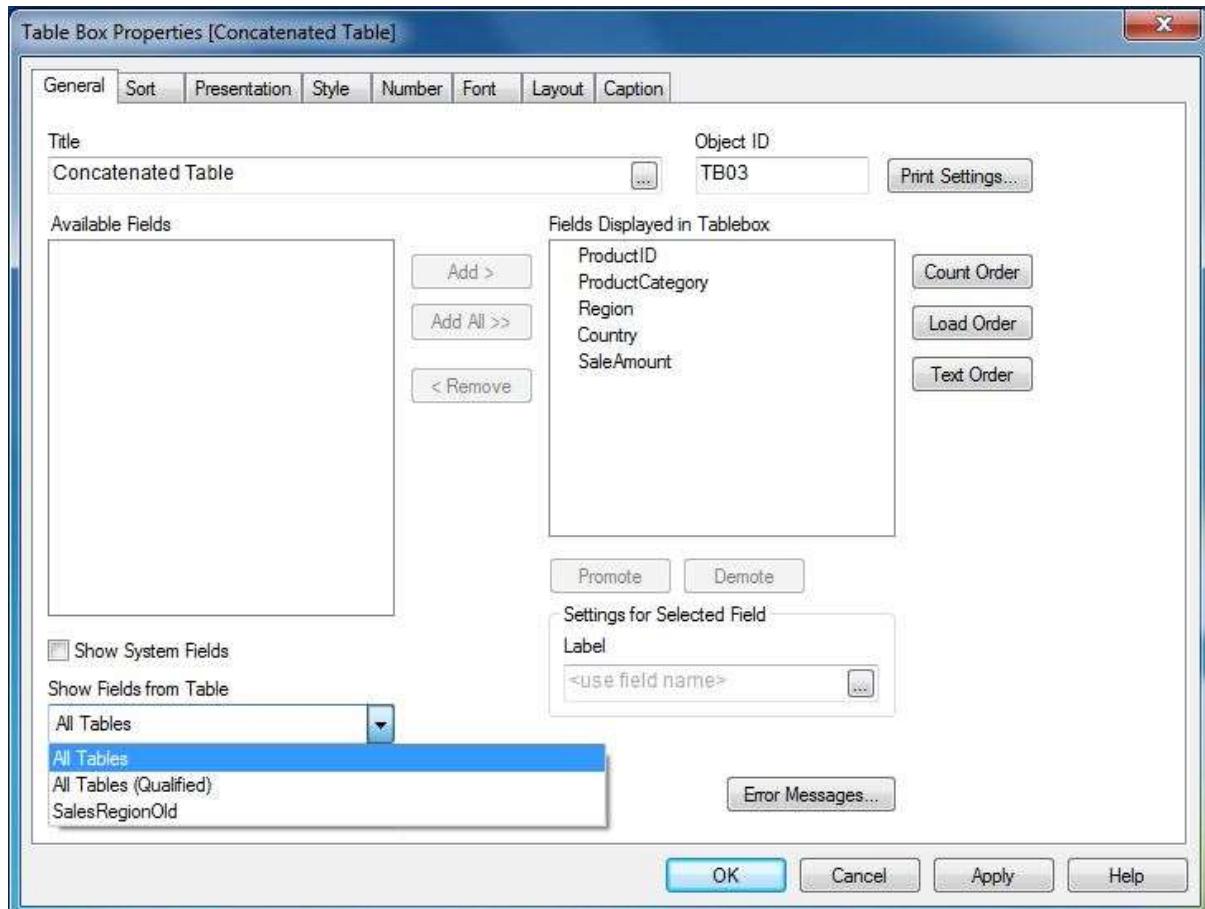
```
SalesRegionNew.csv  
ProductID,ProductCategory,Region,Country,SaleAmount  
6,Arts & Entertainment,North America,USA,1245  
7,Hardware,South America,Brazil,456  
8,Home & Garden,South America,Brazil,241  
9,Food,South Asia,Singapore,1247  
10,Home & Garden,South Asia,China,5462  
11,Office Supplies,Australia,Australia,577
```

## Load Script

We load the above input data using the script editor, which is invoked by pressing **Control+E**. Choose the option **Table Files** and browse for the Input file. Then we edit the commands in the script to apply the concatenation between the tables.



Next, we load the above data to QlikView's memory and create a **Table Box** by using the menu **Layout -> New Sheet Objects -> Table Box** where we choose all the available fields to be displayed as shown below.



## Concatenated Data

Completing above steps we get the Table box displayed as shown below. Please note the duplicate rows for the product ID 6 and 7. Concatenate does not eliminate the duplicates.

The screenshot shows the QlikView interface with a table titled "Concatenated Table". The table has columns: Prod..., ProductCategory, Region, Country, and SaleAmount. The data includes several duplicate entries for Product IDs 6 and 7, illustrating that the Concatenate function does not remove duplicates. The table is displayed in a grid format with a light gray background and white text. The QlikView toolbar and menu bar are visible at the top, and the status bar at the bottom shows "For Help, press F1", the date and time "11/22/2015 8:20:28 AM", and the size "13 X 5".

Prod...	ProductCategory	Region	Country	SaleAmount
1	Outdoor Recreation	Europe	-	4579
2	Clothing	Europe	-	4125
3	Costumes & Accessories	South Asia	-	6521
4	Athletics	South Asia	-	4125
5	Personal Care	Australia	-	5124
6	Arts & Entertainment	North America	USA	1245
6	Arts & Entertainment	North America	-	1245
7	Hardware	South America	Brazil	456
7	Hardware	South America	-	456
8	Home & Garden	South America	Brazil	241
9	Food	South Asia	Singapore	1247
10	Home & Garden	South Asia	China	5462
11	Office Supplies	Australia	Australia	577

## 46. QlikView – Master Calendar

In QlikView, many times we need to create a calendar reference object, which can be linked to any data set present in QlikView's memory. For example, you have a table that captures the sales amount and sales date but does not store the weekday or quarter, which corresponds to that date. In such a scenario, we create a **Master Calendar** which will supply the additional date fields like Quarter, Day etc. as required by any data set.

### Input Data

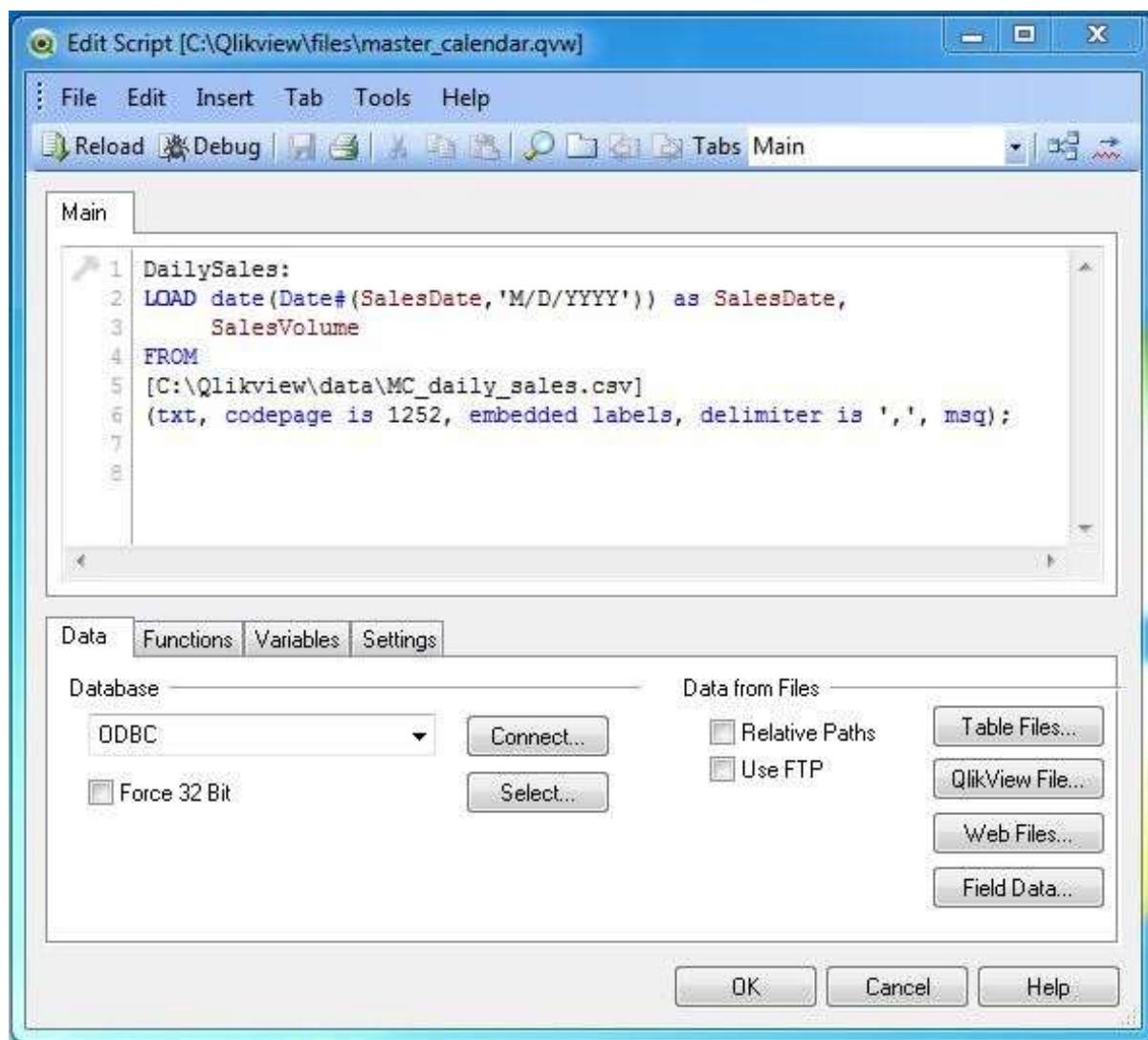
---

Let us consider the following CSV data files, which are used as input for further illustrations.

SalesDate	SalesVolume
3/28/2012	3152
3/30/2012	2458
3/31/2012	4105
4/8/2012	6245
4/10/2012	5816
4/11/2012	3522

## Load Script

We load the above input data using the script editor, which is invoked by pressing **Control+E**. Choose the option **Table Files** and browse for the Input file.



Next, we load the above data to QlikView's memory and create a **Table Box** by using the menu **Layout -> New Sheet Objects -> Table Box** where we choose all the available fields to be displayed as shown below.

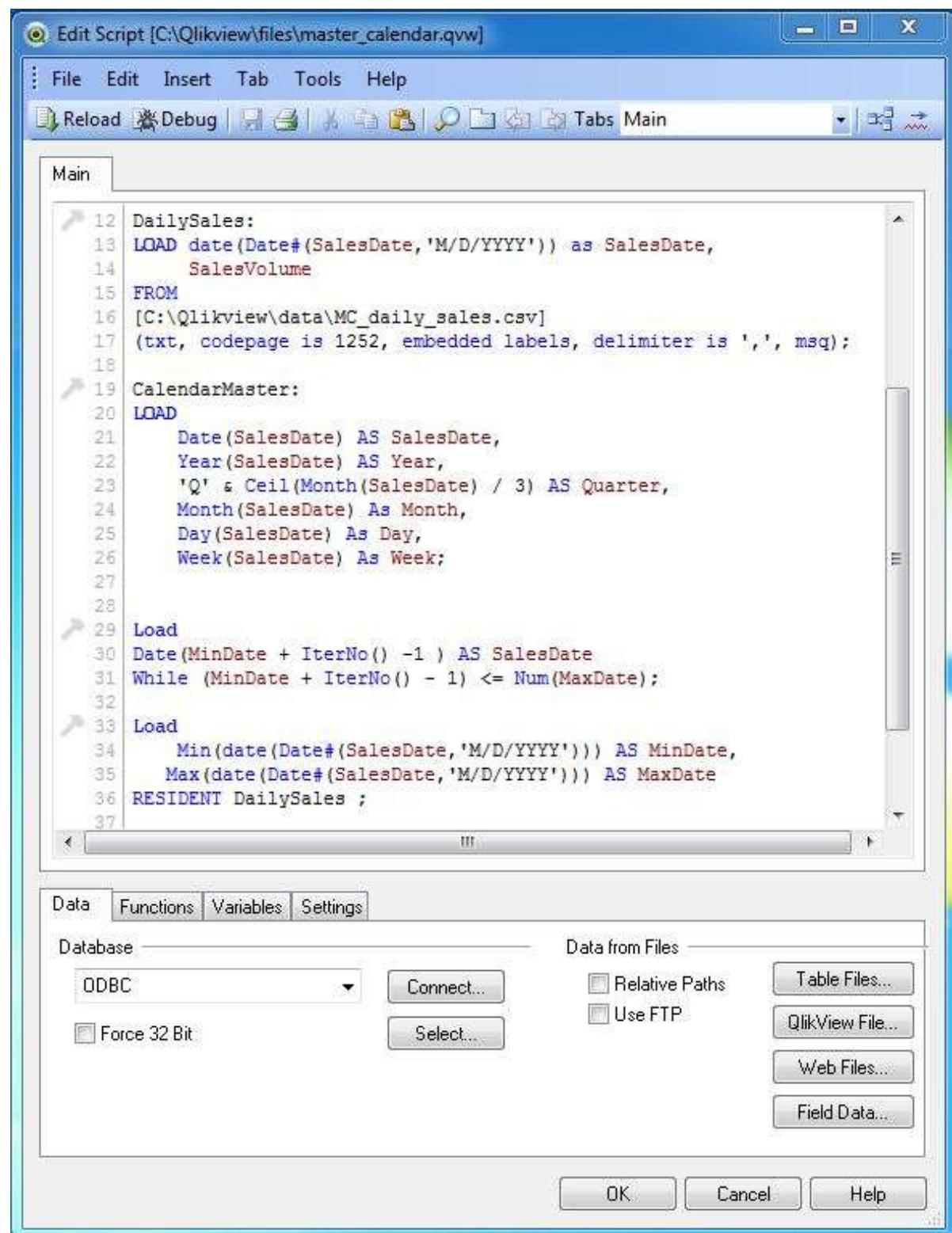
The screenshot shows the QlikView x64 Personal Edition interface. The title bar reads "QlikView x64 Personal Edition - [C:\Qlikview\files\master\_calendar.qvw]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. Below the menu is a toolbar with various icons. The main workspace is titled "Main" and contains a table box. The table has two columns: "SalesDate" and "SalesVolume". The data is as follows:

SalesDate	SalesVolume
3/28/2012	3152
3/30/2012	2458
3/31/2012	4105
4/8/2012	6245
4/10/2012	5816
4/11/2012	3522

At the bottom left of the workspace, it says "For Help, press F1". At the bottom right, it shows the date and time: "11/10/2015 12:36:44 PM".

## Create Master Calendar

Next, we create the Master Calendar by writing the following script in the script editor. Here we use the table DailySales as a resident table from which we capture the Maximum and Minimum dates. We load each of the dates within this range using the second load statement above the resident load. Finally, we have a third load statement, which extracts the year, quarter, month etc. from the SalesDate values.



```

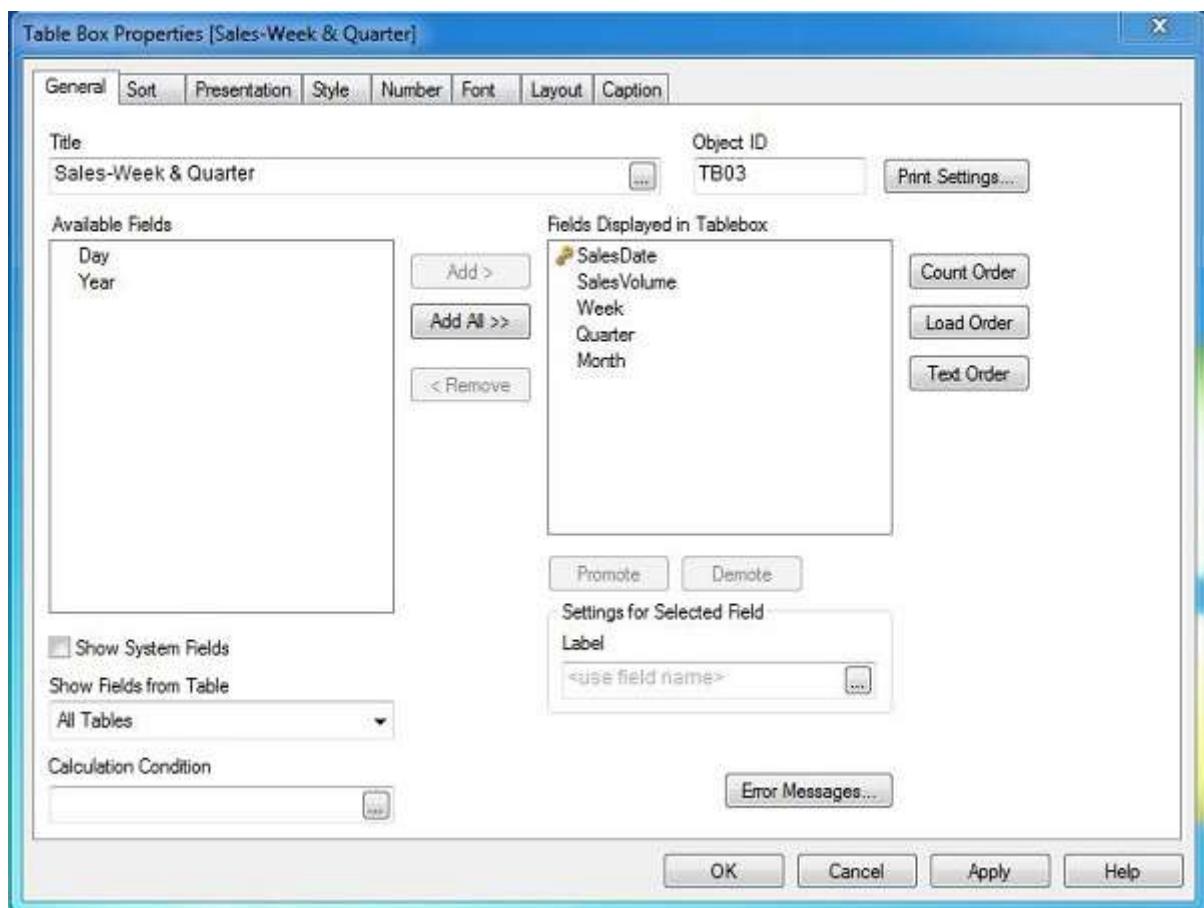
12 DailySales:
13 LOAD date(Date#(SalesDate,'M/D/YYYY')) as SalesDate,
14     SalesVolume
15 FROM
16 [C:\Qlikview\data\MC_daily_sales.csv]
17 (txt, codepage is 1252, embedded labels, delimiter is ',', msg);
18
19 CalendarMaster:
20 LOAD
21     Date(SalesDate) AS SalesDate,
22     Year(SalesDate) AS Year,
23     'Q' & Ceil(Month(SalesDate) / 3) AS Quarter,
24     Month(SalesDate) AS Month,
25     Day(SalesDate) AS Day,
26     Week(SalesDate) AS Week;
27
28
29 Load
30 Date(MinDate + IterNo() - 1) AS SalesDate
31 While (MinDate + IterNo() - 1) <= Num(MaxDate);
32
33 Load
34     Min(date(Date#(SalesDate,'M/D/YYYY'))) AS MinDate,
35     Max(date(Date#(SalesDate,'M/D/YYYY'))) AS MaxDate
36 RESIDENT DailySales ;
37

```

The screenshot shows the QlikView Script Editor window. The title bar says "Edit Script [C:\Qlikview\files\master\_calendar.qvw]". The menu bar includes File, Edit, Insert, Tab, Tools, Help. The toolbar has icons for Reload, Debug, and various file operations. The tabs at the top are set to "Main". The main area contains the script code for creating a Master Calendar. The "Data" tab is selected in the bottom navigation bar, showing database settings for ODBC and options for connecting to files or web sources. Buttons for OK, Cancel, and Help are at the bottom right.

## Select Fields

After creation of the complete load script along with the master calendar, we create a table box to view the data using the menu **Layout -> New Sheet Objects -> Table Box**



## Final Data

The final output shows the table showing the Quarter and Month values, which are created using the Sales data and Master Calendar.

The screenshot shows the QlikView interface with a table titled "Sales-Week & Quarter". The table has four columns: SalesDate, SalesVolume, Week, and Quarter/Month. The data shows sales for the first two weeks of April 2012.

SalesDate	SalesVolume	Week	Quarter Month
3/28/2012	3152	13	Q1 Mar
3/29/2012	-	13	Q1 Mar
3/30/2012	2458	13	Q1 Mar
3/31/2012	4105	13	Q1 Mar
4/1/2012	-	13	Q2 Apr
4/2/2012	-	14	Q2 Apr
4/3/2012	-	14	Q2 Apr
4/4/2012	-	14	Q2 Apr
4/5/2012	-	14	Q2 Apr
4/6/2012	-	14	Q2 Apr
4/7/2012	-	14	Q2 Apr
4/8/2012	6245	14	Q2 Apr
4/9/2012	-	15	Q2 Apr
4/10/2012	5816	15	Q2 Apr
4/11/2012	3522	15	Q2 Apr

For Help, press F1      11/10/2015 12:22:18 PM      15 X 5

## 47. QlikView – Mapping tables

Mapping table is a table, which is created to map the column values between two tables. It is also called a Lookup table, which is only used to look for a related value from some other table.

### Input Data

Let us consider the following input data file, which represents the sales values in different regions.

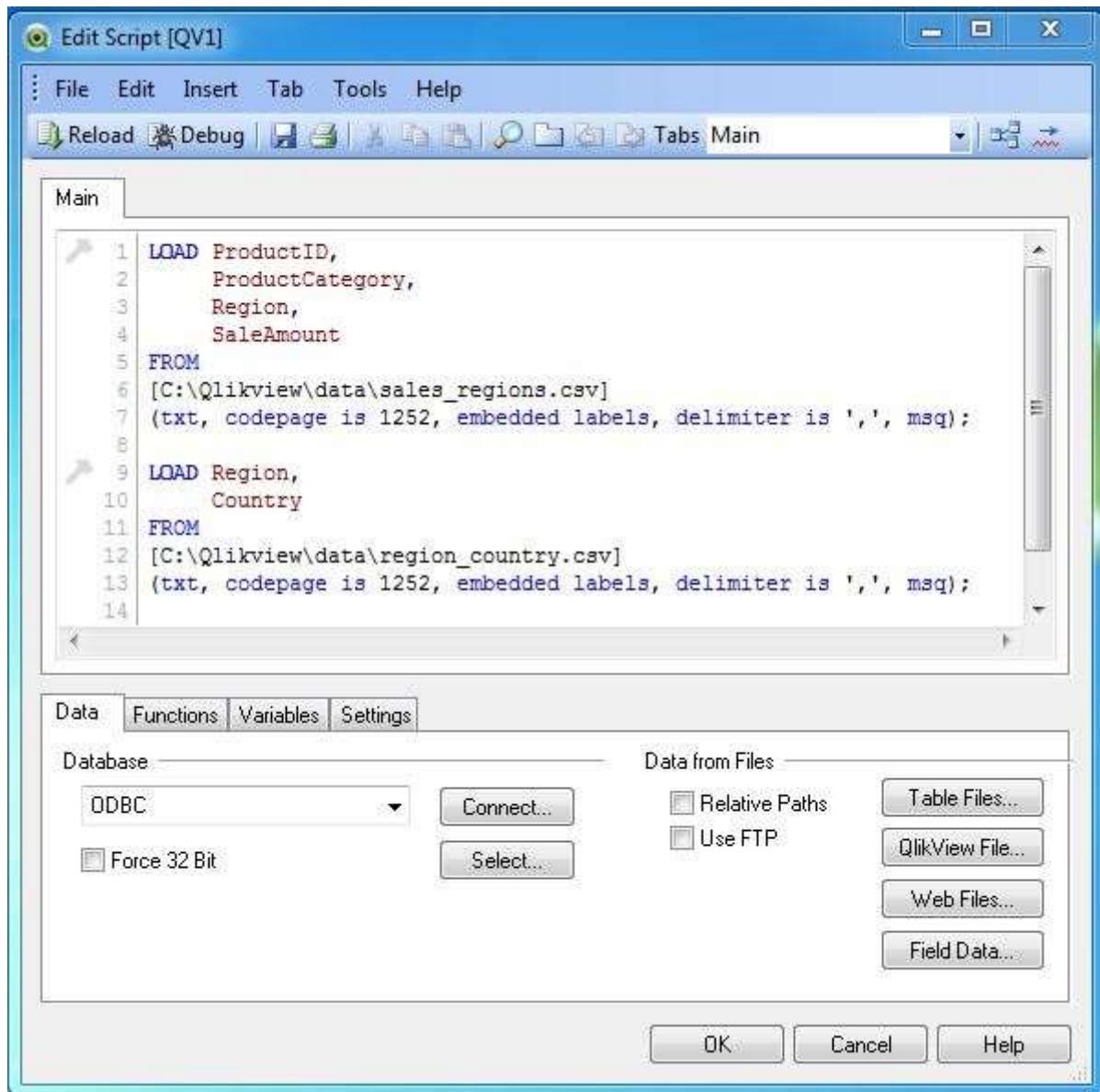
```
ProductID,ProductCategory,Region,SaleAmount
1,Outdoor Recreation,Europe,4579
2,Clothing,Europe,4125
3,Costumes & Accessories,South Asia,6521
4,Athletics,South Asia,4125
5,Personal Care,Australia,5124
6,Arts & Entertainment,North America,1245
7,Hardware,South America,456
8,Home & Garden,South America,241
9,Food,South Asia,1247
10,Home & Garden,South Asia,5462
11,Office Supplies,Australia,577
```

The following data represents the countries and their regions.

```
Region,Country
Europe,Germany
Europe,Italy
South Asia,Singapore
South Asia,Korea
North America,USA
South America,Brazil
South America,Peru
South Asia,China
South Asia,Sri Lanka
```

## Load Script

The above data is loaded to QlikView memory by using the script editor. Open the Script editor from the File menu or press **Control+E**. Choose the **Table Files** option from the **Data from Files** tab and browse for the file containing the above data. Click **OK** and press **Control+R** to load the data into the QlikView's memory.



## Create Table Box

Let us create two table boxes for each of the above table as shown below. Here we cannot get the value of country in the Sales region report.

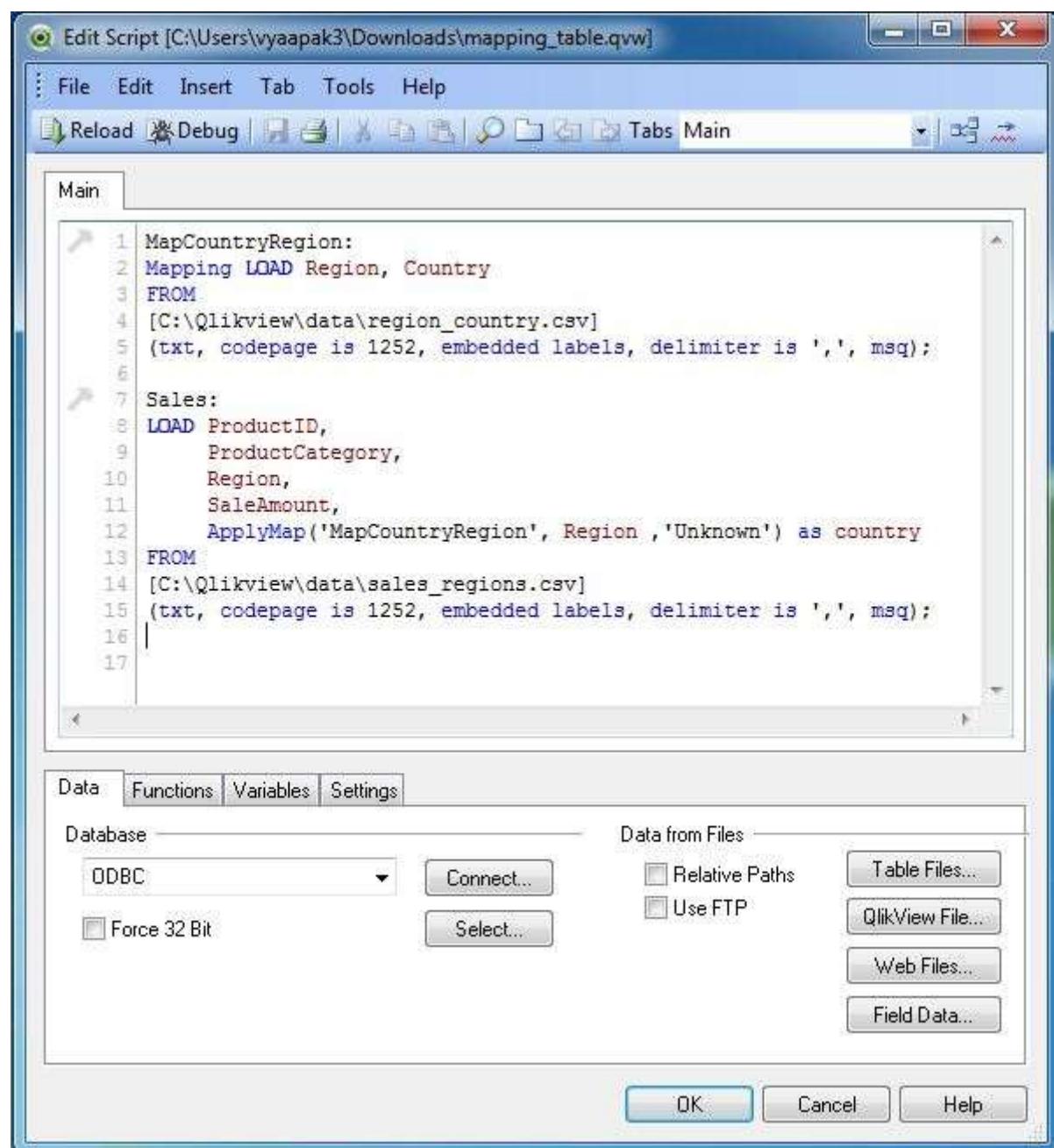
The screenshot shows the QlikView interface with the title bar "QlikView x64 Personal Edition - [C:\Users\vyapak3\Downloads\mapping\_table.qvw]". The menu bar includes File, Edit, View, Selections, Layout, Settings, Bookmarks, Reports, Tools, Object, Window, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and search. The main area is titled "Main" and contains two table boxes. The first table box, titled "Sales regions", has columns: ProductID, ProductCategory, Region, and SaleAmount. The second table box, titled "Country Region", has columns: Region and Country. Both tables have "XL" and "X" buttons in their top right corners.

Sales regions			
ProductID	ProductCategory	Region	SaleAmount
1	Outdoor Recreation	Europe	4579
2	Clothing	Europe	4125
3	Costumes & Accessories	South Asia	6521
4	Athletics	South Asia	4125
5	Personal Care	Australia	5124
6	Arts & Entertainment	North AMerica	1245
7	Hardware	South America	456
8	Home & Garden	South America	241
9	Food	South Asia	1247
10	Home & Garden	South Asia	5462
11	Office Supplies	Australia	577

Country Region	
Region	Country
Australia	-
Europe	Germany
Europe	Italy
North AMerica	USA
South America	Brazil
South America	Peru
South Asia	China
South Asia	Korea
South Asia	Singapore
South Asia	Sri Lanka

## Create the Mapping Table

The following script produces the mapping table, which maps the region value from the sales table with the country value from the MapCountryRegion table.



The screenshot shows the 'Edit Script' dialog in QlikView. The main pane displays the following QlikView script:

```
1 MapCountryRegion:  
2 Mapping LOAD Region, Country  
3 FROM  
4 [C:\Qlikview\data\region_country.csv]  
5 (txt, codepage is 1252, embedded labels, delimiter is ',', msq);  
6  
7 Sales:  
8 LOAD ProductID,  
9     ProductCategory,  
10    Region,  
11    SaleAmount,  
12    ApplyMap('MapCountryRegion', Region , 'Unknown') as country  
13 FROM  
14 [C:\Qlikview\data\sales_regions.csv]  
15 (txt, codepage is 1252, embedded labels, delimiter is ',', msq);  
16  
17
```

Below the script, there are tabs for Data, Functions, Variables, and Settings. The Data tab is selected, showing options for connecting to a Database (ODBC, Force 32 Bit) and selecting Data from Files (Table Files..., QlikView File..., Web Files..., Field Data...). At the bottom are OK, Cancel, and Help buttons.

## Table Chart

On completing the above steps and creating a Table box to view the data, we get the country columns along with other columns from Sales table.

The screenshot shows the QlikView interface with a table titled "Sales Region & Country". The table has the following data:

Prod...	ProductCategory	Region	country	SaleAmount
1	Outdoor Recreation	Europe	Germany	4579
2	Clothing	Europe	Germany	4125
3	Costumes & Accessories	South Asia	Singapore	6521
4	Athletics	South Asia	Singapore	4125
5	Personal Care	Australia	Unknown	5124
6	Arts & Entertainment	North AMerica	USA	1245
7	Hardware	South America	Brazil	456
8	Home & Garden	South America	Brazil	241
9	Food	South Asia	Singapore	1247
10	Home & Garden	South Asia	Singapore	5462
11	Office Supplies	Australia	Unknown	577

## 48. QlikView – Circular Reference

**Circular Reference** occurs when we can traverse from one table to another using two or more different paths. This means you can join Table1 with Table2 directly using a column or you can also first join Table1 with Table3 and then table3 with Table2. This can lead to incorrect result in the output formed by a data model, which loads all these three tables. QlikView prevents the load of such data into its memory once it recognizes a circular reference.

### Input Data

Let us consider the following three CSV data files, which are used as input for further illustrations.

SalesCountries:

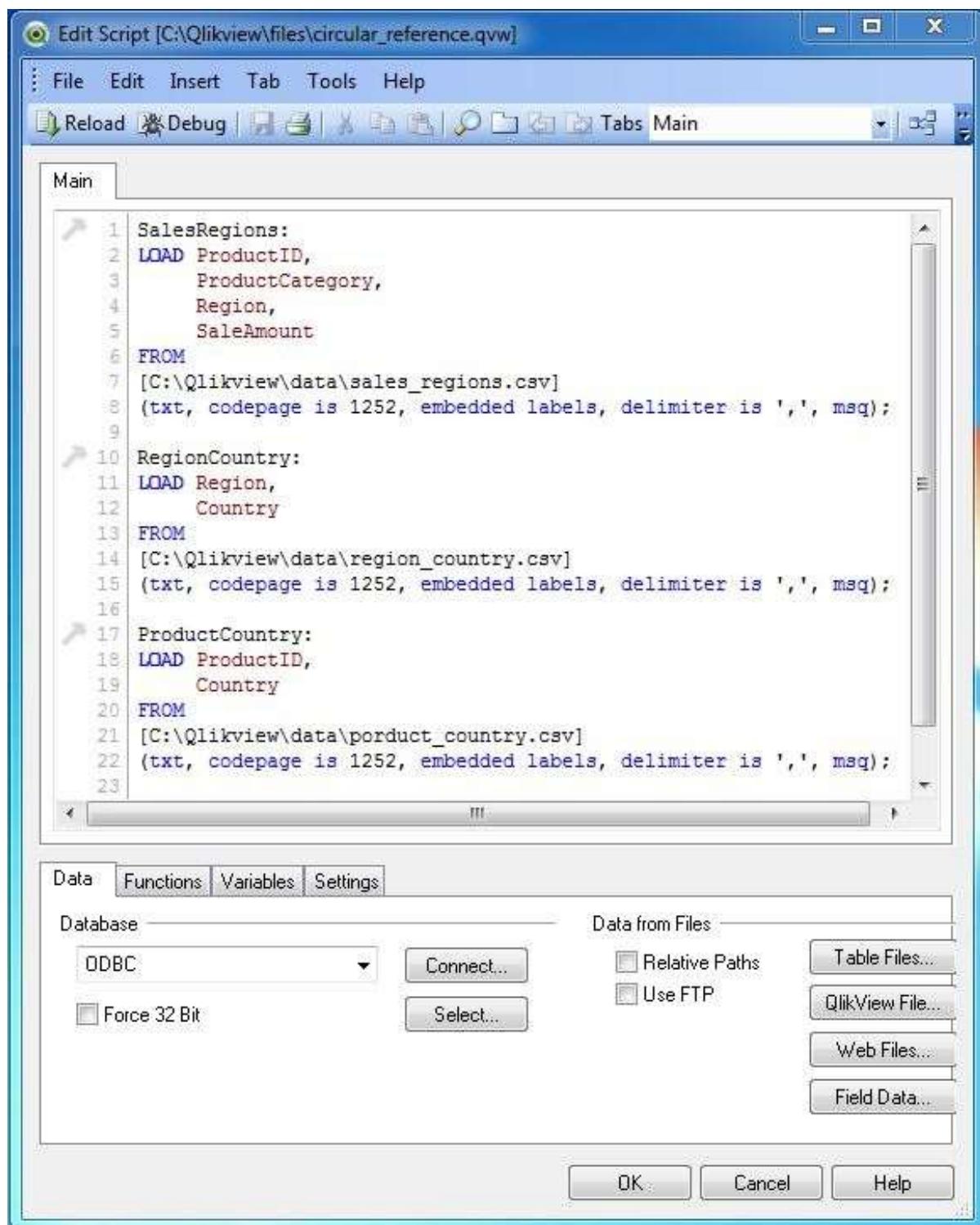
```
ProductID,ProductCategory,Country,SaleAmount  
1,Outdoor Recreation,Italy,4579  
2,Clothing,USA,4125  
3,Costumes & Accessories,South Korea,6521  
4,Athletics,Japan,4125  
5,Personal Care,Brazil,5124  
6,Arts & Entertainment,China,1245  
7,Hardware,South America,456  
8,Home & Garden,Peru,241  
9,Food,India,1247  
10,Home & Garden,Singapore,5462  
11,Office Supplies,Hungary,577
```

ProductCountry:

```
ProductID, Country  
3,Brazil  
3,China  
2,Korea  
1,USA  
2,Singapore  
7,Sri Lanka  
1,Italy
```

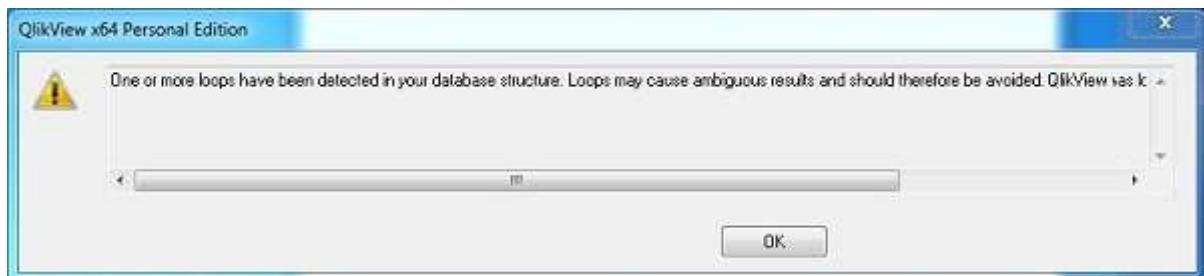
## Load Script

We load the above input data using the script editor, which is invoked by pressing **Control+E**. Choose the option **Table Files** and browse for the Input file.



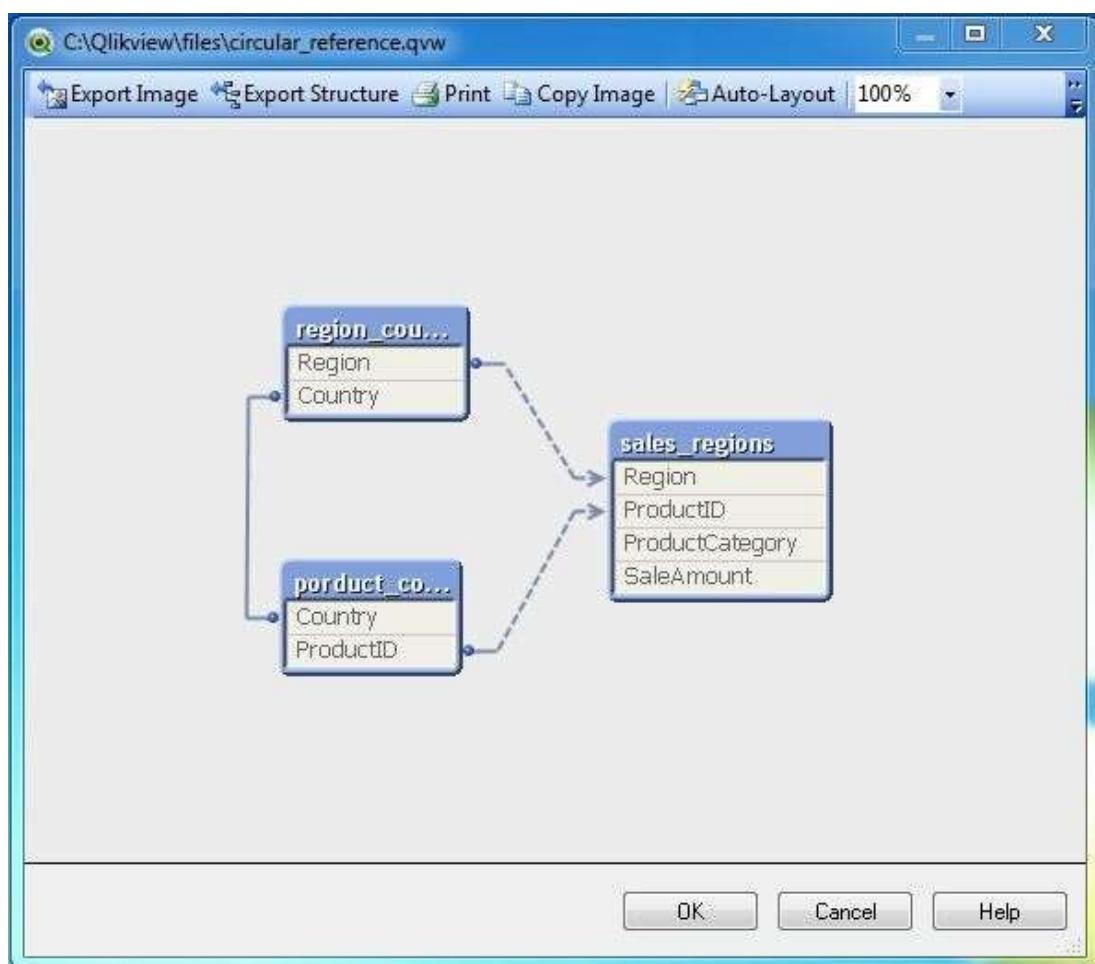
## Data Load

After creating the above script, we load the data to QlikView's memory using the command **Control+R**. This is when we get the error prompt mentioning the presence of circular loop in the tables getting loaded.



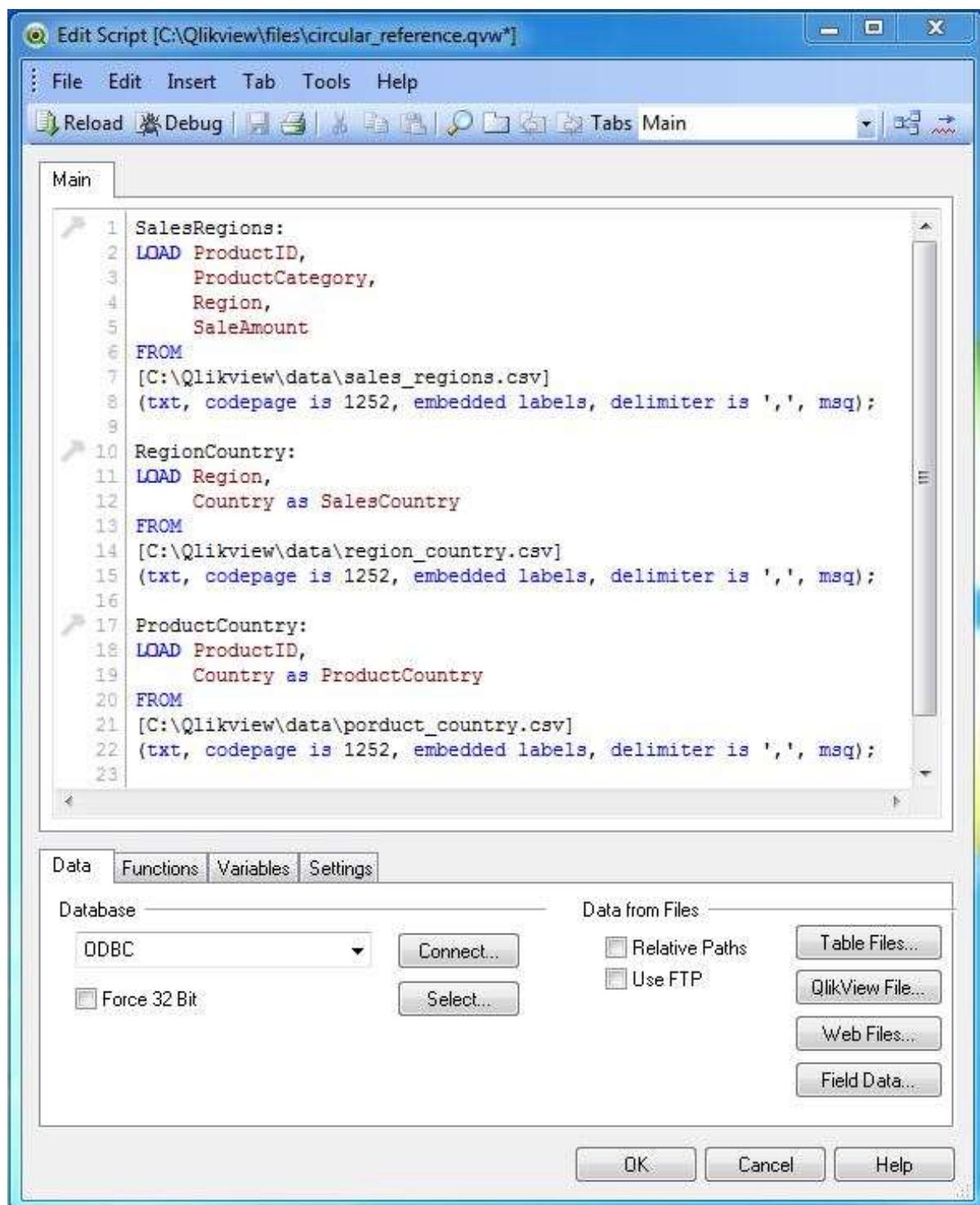
## Data Model

To find the exact cause of the above warning we can look at the data model by using the menu command for table viewer - **Control+T**. The following screen comes up, which clearly shows the circular reference. Here the join between RegionCountry and SalesRegion can be directly achieved using the field **Region**. It can also be achieved by first going to the table ProductCountry, using the field **Country** and then mapping ProductID with Salesregion.



## Resolving Circular Reference

The above circular reference can be resolved by renaming some of the columns in the data sets so that QlikView does not form an association between the tables automatically using the column names. For this, we will rename country column in RegionCountry to SalesCountry. In the data set ProdcuCountry, we rename the Country column to ProductCountry.



The screenshot shows the 'Edit Script' dialog in QlikView. The title bar says 'Edit Script [C:\Qlikview\files\circular\_reference.qvw\*]'. The menu bar includes File, Edit, Insert, Tab, Tools, Help. The toolbar has icons for Reload, Debug, and various file operations. The main area is titled 'Main' and contains the following QlikView script:

```

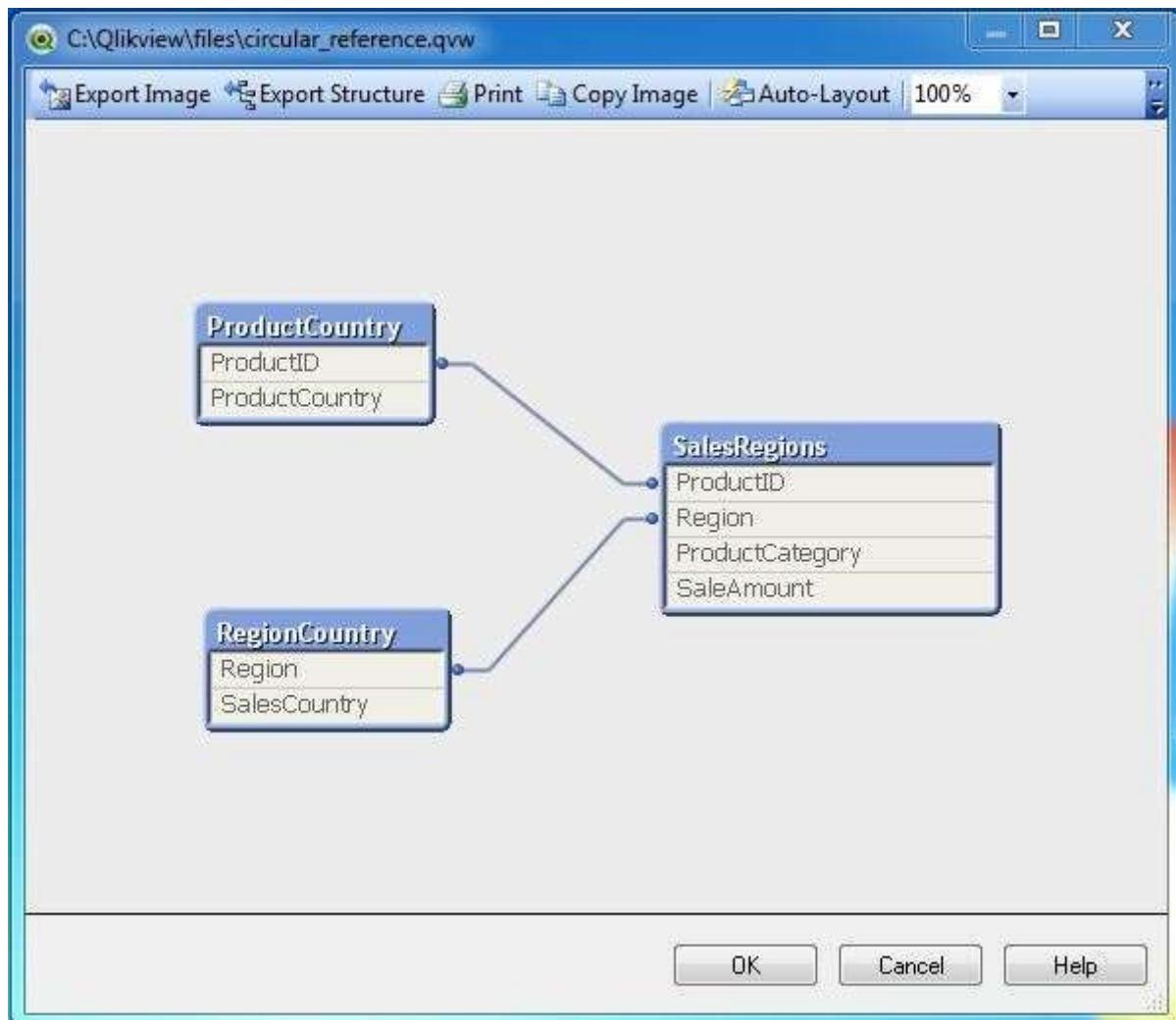
1 SalesRegions:
2 LOAD ProductID,
3     ProductCategory,
4     Region,
5     SaleAmount
6 FROM
7 [C:\Qlikview\data\sales_regions.csv]
8 (txt, codepage is 1252, embedded labels, delimiter is ',', msg);
9
10 RegionCountry:
11 LOAD Region,
12     Country as SalesCountry
13 FROM
14 [C:\Qlikview\data\region_country.csv]
15 (txt, codepage is 1252, embedded labels, delimiter is ',', msg);
16
17 ProductCountry:
18 LOAD ProductID,
19     Country as ProductCountry
20 FROM
21 [C:\Qlikview\data\poroduct_country.csv]
22 (txt, codepage is 1252, embedded labels, delimiter is ',', msg);
23

```

Below the script, there are tabs for Data, Functions, Variables, and Settings. Under the Data tab, there are sections for Database (ODBC dropdown, Connect..., Force 32 Bit checkbox) and Data from Files (Relative Paths, Use FTP checkboxes, Table Files..., QlikView File..., Web Files..., Field Data... buttons). At the bottom are OK, Cancel, and Help buttons.

## Rectified Data Model

The Rectified data model after renaming the column above can be seen using the command **Control+T**. Now we can see that the relationship between the tables does not form a loop.



Pressing **Control+R** to reload the data does not give us the warning anymore and we can use this data to create reports.