TIBCO Spotfire – A Comprehensive Primer

The need for agile business intelligence (BI) is growing daily, and TIBCO Spotfire combines a self-service feature set with essential enterprise governance capabilities to deliver analytics solutions. Spotfire is easy and intuitive to use and can also be a rewarding development environment for all BI users.

Starting with data and visualization concepts, this book takes you through increasingly advanced topics toward the goal of becoming a professional analytics solution provider. It explains data and visualization concepts and follows through with practical examples, such as getting data into Spotfire and using its visualization features. You will learn to build sophisticated user interactivity frameworks and spatial analytics, and administer the Spotfire environment.

This book ends with a case study on self-service BI and leaves you equipped to contribute to the demand for more agile BI.

Who this book is written for
If you are a business user or data professional, this book will give you a solid grounding in the use of TIBCO Spotfire. This book assumes no prior knowledge of Spotfire or even basic data and visualization concepts.

What you will learn from this book
- Master core data concepts, model the source data, slice and dice dimensional data, and define metrics and key performance indicators
- Learn to deploy Spotfire’s suite of visualization types effectively and intelligently
- Build user-friendly analytics frameworks with Spotfire text areas and property controls
- Understand the different ways Spotfire can be deployed, licensed, and administered
- Use Spotfire’s location analytics capabilities to create interactive geographic and non-geographic spatial analyses
- Write IronPython scripts against the Spotfire API to tackle and solve challenging analysis framework requirements
- Combine all your newfound or enhanced data, visualization, and programming skills to build self-service business intelligence solutions
- Explore Spotfire’s predictive analytics, mobile metrics, event analytics, and advanced data services capabilities

Create innovative enterprise-class informatics solutions using TIBCO Spotfire
In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 “Show Me the Data”
- A synopsis of the book’s content
- More information on TIBCO Spotfire – A Comprehensive Primer

About the Author

Michael Phillips has worked with TIBCO Spotfire on a daily basis for the past 3 years, designing and building solutions for complex clinical informatics challenges. He has a strong mix of scientific, business, and technical experience. Having gained a PhD in biochemistry, Michael worked as a general science and medicine editor for 10 years, and also as an IT manager for 15 years, specializing over time in business intelligence (almost 10 years now). He currently works as a clinical informatics product manager in a dedicated innovation team at ICON plc, a large global clinical research organization with a very strong product offering in clinical informatics. His work spans a comprehensive range of activities from business, commercial, and scientific analysis and strategy right through to the technical work of building solutions in Spotfire.

He has a passion for business intelligence, particularly data visualization and self-service data exploration. Although his IT experience is wide-ranging and he is very comfortable with technology, he retains a strong business focus and believes that self-service informatics is a difficult but absolutely essential challenge in today's culture of "big data" and information on demand.

Many thanks to my friends and family for their support and encouragement.
TIBCO Spotfire – A Comprehensive Primer

Big data is not new, nor are data visualization and analysis. Around 5,000 years ago, the Babylonians were using a symbolic number system and mathematical tables not only to record and analyze mundane market information, but also to record and process astronomical data to reach conclusions about the nature of the cosmos.

Technology has advanced considerably since the clay tablets of Babylonian times, and more data is available then ever before, but data volume and complexity and the techniques available to handle data are relative concepts. It has always been challenging to derive insight and meaning from data, and it always will be.

TIBCO Spotfire is one of a handful of general-purpose analytics platforms that bring data integration, transformation, analytics, and visualization together in a single, enterprise-class development environment. The rich feature set and enterprise architecture allow you to create simple, centralized dashboards from spreadsheets or develop sophisticated self-service business intelligence frameworks that integrate multiple "big data" sources or model specialized and varied data through advanced analytics algorithms to develop and test new informatics hypotheses.

I have been working with Spotfire since 2011 to do all of the above in the field of clinical research, where good data analysis and insight lead to healthier and longer lives. I wrote this book to share my experience with you and give you a solid grounding in the use of this amazing analytics product. I deliberately avoided using any examples from my clinical background and instead chose to use more general, and I hope interesting, examples to illustrate Spotfire's many capabilities.

Whether you want to do mundane business analysis or push the boundaries of data science, this book is the beginning of your journey into Spotfire's panoramic analytics landscape. Fasten your seatbelt if you wish but keep this chart at hand at all times!

What This Book Covers

Chapter 1, Show Me the Data, covers data access, starting with an Excel spreadsheet and moving on to other data sources.

Chapter 2, Visualize This!, gets you to think about the visualization of information and how to do this with Spotfire.

Chapter 3, Analyze That!, shows you how to quickly transform even simple analysis into professional and powerful analytical toolkits through the addition of context and input controls.
Chapter 4, The Big Wide World of Spotfire, looks at the main Spotfire platform components and covers basic platform administration.

Chapter 5, Source Data is Never Enough, shows you how to use Spotfire's data tools to transform your source data, when necessary, into a form that is more suitable for your analysis needs.

Chapter 6, The World is Your Visualization, covers visualization types and advanced visualization features not discussed in Chapter 2, Visualize This!

Chapter 7, What's Your Location?, shows you how to use Spotfire's location analytics capabilities to put geographic and other spatial context to your analysis.

Chapter 8, The Secret Life of Python, demystifies the Spotfire API and shows you how to use IronPython to program visualization properties, data properties, and more.

Chapter 9, It's All About Self-service These Days, covers the topic of guided analysis and presents a case study in the delivery of self-service business intelligence frameworks.

Chapter 10, Beyond the Horizon, explores some optional and advanced Spotfire components not covered elsewhere in this book: TIBCO Enterprise Runtime for R, JavaScript, mobile metrics, event analytics, and advanced data services.
when you start spotfire for the first time, your first task is to load some data. this data can come from a file, a database, or even the clipboard. this chapter will show you how to get started quickly with a microsoft excel spreadsheet and move on to work with other data sources.

data is at the heart of all analysis, and it's important that you know, not only how to load data into spotfire, but also how data works. if you handle a lot of data in spreadsheet form, you will no doubt understand its content and meaning very well. you might even have developed advanced and insightful representations of your data. however, there is so much more you can do with spotfire to improve the handling of this subject matter.

importing data into spotfire is just the beginning. to progress into its rich analytic world, you will have to become familiar with the relational database model. you will have to learn some formal data concepts. we will therefore spend some time taking a look at some basic database principles to set you on your way to advance quickly beyond the limited world of the spreadsheet.

at the end of the chapter, you will have built a solid foundation to work through the more visually exciting tutorials in this book, and indeed, for your future use of spotfire as an everyday analysis tool. what's more, you'll be very pleasantly surprised at how easy it is to get started.

before we begin, please read the preface if you haven't already done so. it contains some important prerequisites for the examples we will cover in this and later chapters. this chapter does not go into any detail about the spotfire platform, focusing instead on the manipulation and analysis of data. you should find this a more rewarding and productive starting point. we will cover the tibco spotfire platform in detail in chapter 4, the big wide world of spotfire.
In this chapter, we will cover the following topics:

- Adding data to Spotfire from files
- Key data concept—basic row/column structure of a data table
- Key data concept—date types
- Using the inbuilt filters in Spotfire
- Key data concept—data relationships
- Linking data in Spotfire
- Connecting directly to a database
- Key data concept—data queries

**Adding data to Spotfire from files**

There are usually several ways to achieve the same result in Spotfire and, as you learn to use the tool, you will develop your own preferences. When you launch Spotfire, you will see all the main options to load data.

From the opening menu, you can use the **Open File** icon or the **Add Data Tables** icon. If you look at the main menu bar, you’ll notice the **File** option. If you select this option, you should see the **Open** and **Add Data Tables...** options. Similarly, if you look at the icon bar just below the main menu bar, you’ll notice symbols to open a file and add data tables. Hover your mouse over the icons and you will get a description of their functions.
Whether you want to use the menu options or the icons is a personal preference. The difference between opening a file and adding a data table is important, however.

Opening a file means opening a datafile or a saved Spotfire analysis file and closing any open Spotfire file in the process. Spotfire will prompt you to save your file before you open the new file.

Adding a data table means adding data content to an open Spotfire analysis file, keeping all its existing content. If you simply launched Spotfire, opening a file and adding a data table amount to the same thing but, if you want to add additional data tables, you must use the add data table option.

**Importing a Microsoft Excel spreadsheet into Spotfire**

Let's open a Microsoft Excel spreadsheet in Spotfire.

The data used is BaseballPlayerData.xls, which you can download from http://www.insidespotfire.com or copy from the TIBCO Spotfire professional client installation directory (~TIBCO\Spotfire\#.\#\Example Data\Baseball) on your PC. The TIBCO file is in text form, so you will need to convert it to Microsoft Excel first.

1. Start by clicking on the folder icon (on the far left in the previous screenshot).
2. You will be presented with a standard Open file dialog, allowing you to navigate to the spreadsheet file.
3. Spotfire will open a dialog window asking you to confirm or change key aspects of the Microsoft Excel file.

4. The first thing to notice is the **Worksheet** selection dropdown at the very top of the dialog window. Spotfire can only import one worksheet at a time. There is only one sheet in our file, so we can ignore this option.

5. The next thing to notice is the preview of your data and its structure. Spotfire will automatically detect and assign column headers and data types, but you can change any of these settings. You can also tell Spotfire not to import specific columns or rows.

6. We want to open the file with all defaults, so we're just going to click on **OK**, but please do explore the dropdown options for columns and rows and experiment with the settings. The core philosophy of Spotfire is discovery, so start as you mean to continue and explore all the options.

7. Once you click on **OK**, Spotfire will import the spreadsheet, create a new page in your analysis, and display the data as a visualization. The type of visualization will depend on the default option you set under **Tools | Options | Document**.

8. If the data is not displayed in tabular form, then first close the visualization by clicking on **X** in the upper right-hand corner and then create a **Table visualization** by clicking on the New Table icon in the Spotfire toolbar.
After you import the data and set up the Table visualization, you should see the following:

Take a look at the General tab in Data Table Properties, which you will find under the main Edit menu. The default Store data option for the table you loaded is Linked to source. This option means that the data always remains in the source file and is pulled into memory by Spotfire when the analysis file is opened or the data is refreshed (using the Refresh Data button next to the table list). To make the analysis file more portable, you can change the status to Embedded in analysis, which means that the data resides in the analysis independent of the original file. Refresh Data still works and updates the embedded data with any changes made in the source file.

9. Save your analysis file by clicking on the disk icon or by selecting Save or Save As in the File dropdown list. Name it BaseballPlayerData. We will be returning to this file in future examples.
Importing a text file into Spotfire

Ready for another example? This time, we're going to import a text file into Spotfire, which provides some useful options for structuring text files into a more analyzable form.

1. Follow the exact same procedure as for the Microsoft Excel spreadsheet, except this time we are going to open the data from a text file (BaseballPlayerData.txt, which you can download from http://www.insidespotfire.com).

2. Once again, you will be presented with an import dialog and data preview, but this time you will be able to customize how the file is delimited, which means how you want to separate the information into columns. You can also change properties related to text encoding.

3. Spotfire has correctly detected the tab separator in the text. Simply accept the defaults and click on OK to import the text into Spotfire.

The data will display exactly as it did with the Excel import. Compare Spotfire's output with the raw file opened in Notepad.
Importing other file types into Spotfire

Spotfire can import data from other structured file types, such as Microsoft Access (.mdb) and SAS (.sas). The basic process to open these files is no different than it is for Microsoft Excel and text, except Spotfire uses the data structure embedded and defined in these files and gives you control over which parts of the data to import.

Below is an example dialog for a SAS file. The three columns we don’t want to import are highlighted. The next step is to click on the < Remove button, after which the columns will flip to the left-hand window. When we’re happy with the selections, we click on OK, and Spotfire will import a dataset based on the selections.

Key data concept – basic row/column structure of a data table

If you already understand the difference between rows and columns, great! You can skip this section. If you are not sure, then read on because your understanding of this concept is essential for data analysis and report authoring.

The columns in the dataset represent how the information has been categorized. They exist even if there is no data. Most people these days are familiar with Microsoft Excel. When you start a new spreadsheet, one of the first things you probably do is decide what types of information you are going to add; for example, First Name, Last Name, and Department in a simple human resources spreadsheet.
Once you have structured your spreadsheet in this way with column headings, you can begin to add the actual information, row by row. Your columns don't change in number or description, but your rows grow and shrink in number, and changes might be made to the information at any time.

One important distinction between data tables and spreadsheet workbooks is the way in which almost everyone manipulates the visual layout of the rows in a spreadsheet. You might, for instance, not repeat a department value until it changes; you might merge cells to improve the look and feel. You cannot do this with data tables. If the department column value for the first three rows is Marketing, then Marketing must be repeated in each row.

Data content (the values in the cells) can change, but it is always filtered and selected through references to the column names. For example, the request, show me all the records for Marketing, might produce zero rows or several million rows, depending on how many records the query finds with the word Marketing in the Department column.

In a spreadsheet, it is easy to build calculations that reference any cell in the matrix. If you want to make a calculation in a data table, such as sale amount - cost amount, you can only do that across each row. You cannot subtract the cost amount in one row from the sale amount in another. This is a key distinction between data tables and spreadsheets. It might seem like a limitation in a data table, but the discipline of that structural integrity ultimately allows you to create very powerful analyses, and there are ways to change the structure of a data table into new forms to support a particular calculation requirement.

Reports and visualizations are built around column names. For example, you might want to create a simple sales by region visualization. What you are doing is putting the sales column against the region column and asking the visualization engine to populate the chart or graph with whatever row values are present beneath those columns in the given data set.
Key data concept – data types

Another key data concept that is important to mention at this point is that of data type. This concept is equally important in spreadsheets; it's just that spreadsheets don't generally force you to declare the data type, and they allow you to mix and match data types under individual columns. In data tables, each column must have a single data type for all the values in all rows.

So what does data type mean? There are, in essence, three types of data:

- Numbers
- Dates
- Text

They are defined as such because they have distinct properties. Numbers can be used in calculations, and the vast array of mathematical functions and operators can be brought to bear on them. Text can be parsed, concatenated, counted, and arranged into categorical hierarchies. Dates have a special meaning and can be used in time calculations and hierarchies (year>month>day, for example).

However, life is never simple, and there are several subtypes of these data types and a few other special data types that you need to understand.

Spotfire uses 12 data types, and all data columns imported into Spotfire must be put into one of these categories. The following table describes the ones you will use the most. Please consult the TIBCO Spotfire documentation for a more technical description.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer</td>
<td>Numbers with no decimal places</td>
</tr>
<tr>
<td>Real</td>
<td>Numbers with decimal places</td>
</tr>
<tr>
<td>Date</td>
<td>Date with no time element</td>
</tr>
<tr>
<td>DateTime</td>
<td>Date and time combined</td>
</tr>
<tr>
<td>String</td>
<td>Text</td>
</tr>
<tr>
<td>Boolean</td>
<td>Logical True or False</td>
</tr>
</tbody>
</table>
Using the inbuilt filters in Spotfire

Spotfire's inbuilt filters offer a very powerful and immediate way to start analyzing your data. Every time you add a data table to an analysis file, Spotfire creates a filter for each column. Just reflect on this for a minute: if we are going to try to filter or screen our data in some way, we have to do so on the basis of the values in one or more of the data table's categories. That is why a filter always corresponds to a table column and its values to whatever data currently populates that column through the rows, or records, in the table.

Let's take a look at the baseball data we loaded earlier in this chapter.

1. Open the Spotfire analysis file you saved earlier (BaseballPlayerData.dxp) or reload the data if necessary.

2. Look at the Filters panel on the right-hand side of the analysis. If, for some reason, there is no Filters panel, simply go to the View menu and select Filters. Filters cannot be deleted in Spotfire. They can be hidden from view and configured in custom ways, but the underlying filters always remain.

3. Each of the filters has a header that describes the associated column. The values under each heading reflect the data found in the table's rows, so the filters can be a useful way to explore the scope of the data. For example, there are only two possible values under League — AL and NL — that's useful to know.
4. You’ll notice that the filters take different forms for the different columns. That’s because Spotfire chooses from a set of six filter types to best suit the data that is found under each column. Some columns have a small set of unique values, such as the League column, all the rows for which have either AL or NL; the check-box filter type is ideal for this scenario. In contrast, the Games Played column is numeric, with values ranging from 86 to 162, so a slider filter is more appropriate.

5. Go to the filter for At Bats and move the left and right sliders until they define a range between 500 and 550. If you double-click on the number, you can type in a value and then simply hit the return or enter key to move the slider precisely.

6. Go to the filter for Runs and move the left slider up to 80.

7. What we’re looking at now is records for players who were at bat between 500 and 550 times and who scored 80 runs or more. If you use the table's horizontal scrollbar, you can move through the columns until you find the Player column to see the players in question.

8. Notice how the other filters have adjusted to reflect the values available for the filtered dataset and how the row status at the bottom left of the analysis window has changed to 12 of 278 rows.
9. Explore the filters yourself to get a feel of how they work and what is possible. To reset all filters back to their default state, simply click on the reset filter icon or select **Reset All Filters** from the **Edit** menu.

An important point to stress here is that we haven't removed any data from the underlying table. Our visualization—in this case, a basic table listing—has changed and *lost* some rows, but as you saw, when you reset the filtering, the visualization adjusted dynamically and displayed the refreshed results. We will cover additional aspects of filtering as the book progresses because it is central to the way Spotfire works.

**Key data concept – data relationships**

Let's quickly review what we learned so far:

- We covered how data tables are structured
- We discussed how data tables differ from spreadsheet workbooks
- We know how to load a data file into Spotfire

Now, we're going to explore one of the most powerful aspects of data tables: linking or relating the information in two or more different tables. Building structural links between data tables is at the heart of the relational database model that underpins well-known databases, such as Oracle and Microsoft SQL Server. Spotfire is not a relational database as such, but it uses relational principles and provides some table relationship functionality.

If you think back to our discussion of rows and columns, you should be able to conclude that the relationships between tables are defined at the column level. In other words, if we have a table with a **Department** column, we might want to relate it to another table that also has a **Department** column. In plain English, we're going to say something like "relate these two tables based on the row values found under their respective Department columns; if you select **Marketing** records from one table, the relationship will be associated with any **Marketing** records in the other table."
An instructive corollary of this relationship—and this is a very important concept for you to grasp—is that the row values must match perfectly but the column names can be completely different. You can define a relationship between a "Department" column in one table and a "Dept" column in another. There is a limit, however, to how much you can tell the relationship engine how to interpret the values. For example, will Marketing, marketing, and MARKETING match? You have to explicitly define such match rules.

How to link data in Spotfire

Let's see how table relations work in practice. First, we have to add some new data.

The data used here is BaseballPlayerData.xls and BaseballTeamData.xls, which you can download from http://www.insidespotfire.com.

1. Open the baseball analysis file you've been working with.
2. We're now going to add a second data table to the analysis, so select Add Data Tables... from the File menu.
3. **BaseballTeamData.xls** is a Microsoft Excel spreadsheet derived from the baseball player dataset and contains team-level information only. We'll import it just like we did in the earlier examples, accepting the column definition defaults as before. The only difference is that, this time, you will add the table to an existing analysis. This datafile has one row for each team and provides aggregated values (mainly averages) for all the individual statistics provided by the player data table.

4. After you add the data table, Spotfire automatically adds a new page with a default visualization.

### How to relate two tables

A core feature of Spotfire is the seamless way in which you can interact with multiple visualizations based on multiple data tables. Be patient, we will move on to the visualization part soon, in *Chapter 2, Visualize This!*. First, you must learn the key technique to link or relate different tables.

We are going to relate the player data table, where each row in the data represents an individual baseball player, with the team data table, where each row represents a baseball team. The obvious link is the team name, so that is how we will relate the two tables. Once related, the single row for, say, team Detroit in table 2 will be linked to the nine rows for team Detroit in table 1.

Data **granularity** is an important concept. One table here has a finer granularity (player) than the other (team). Such tables can be related through the lowest level of granularity that they share—team, in this case.
Table relations are defined in the analysis file's **Data Table Properties**.

1. From the **Edit** menu, select **Data Table Properties**. The dialog that opens shows you all the tables currently loaded into your analysis file.
2. Navigate to the **Relations** tab and click on the **Manage Relations** button and then the **New** button to get to the table relations dialog.

3. Select one table as the left data table and the other as the right table; it doesn't matter which is which. Now you need to decide how these tables are related by selecting the columns through which to connect them. Use the dropdown lists in each case to select **Team** because that is the common link between these two data tables.

4. If you want to define a more flexible matching rule, use the **Left Method** and **Right method** dropdown. For example, you could select **Upper** in both cases. This setting will ensure that the match is case insensitive. The data will not change, and one table might still have **Detroit** and **DETROIT** in any given row, but they will match the relationship because they will be converted to all uppercase for matching purposes.
5. Finally, click on OK in the New Relation dialog, click on OK in the Manage Relations dialog, and click on OK in the Data Table Properties dialog, noting as you do that both tables have been assigned the same color index, indicating a defined relation.

6. You can edit a relation at any time by going back to the Manage Relations dialog, selecting the relation of interest, and clicking on Edit.

7. If you need to relate two tables on the basis of more than one column, simply add each relation pair one by one.

**Column matching**

Another way to relate data in Spotfire is column matching, which is used to determine how multiple tables should be treated when used alongside each other in the same visualization. Spotfire will automatically match columns with the same name, regardless of whether you defined any relationship between the two tables.

You can override Spotfire or add new column matches by performing the following steps.

1. Navigate to the Column Matches tab.
2. You will notice the three columns already matched by Spotfire.
3. You can select and delete any of these columns using the **Delete** button.

4. You can edit any of the matches or you can add a new match. Spotfire will allow you to match any pair as long as they have the same data type. In the screenshot, **Team** and **Position** have been selected as a potential match, which doesn't actually make any sense. It does illustrate how important it is to understand all aspects of your data, not just its structure but also the values.

**Connecting to a database**

Working with text files and spreadsheets can be very convenient but, sooner or later you are going to want to work with databases, which offer many advantages such as scale and centralization. If you want to build scalable, enterprise analytical solutions, you will have to maintain your source data in a database or even distributed across multiple databases. Spotfire provides many ways to connect to databases, and you can retrieve data simultaneously from multiple databases and files.

Up to Version 3 of Spotfire, we had the option to connect to databases such as Oracle, Microsoft SQL Server, and Microsoft Access using standard data connectors such as ODBC, OleDb, Oracle client, and SqlClient. Explaining how these types of connectors work is beyond the scope of this book. You will find detailed explanations on the Internet.

They all require database-specific drivers, which you can think of as like an operating system plugin. Many of these drivers come bundled with operating systems; some you will have to download and install. Again, you will find lots of help on the Internet. The best place to look is on the websites of the various database vendors.

They all require some form of **connection string**, which is just a way to tell the driver the address of your database and the credentials you are using to connect. Most databases will require authentication before you can connect and access their data. The website [http://www.connectionstrings.com](http://www.connectionstrings.com) is a good starting place. Sometimes the entire connection string is pasted as a literal string into a single configuration box; sometimes the individual elements of the connection string are entered into fields in configuration form; it just depends on the connector.

**Using ODBC**

**Open Database Connectivity (ODBC)** is one of the easiest drivers to use. It might not be perfect for all applications, but it's usually a reliable way to get started. As you gain experience, you will use other options. It might also depend on the database(s) you wish to connect to.
There are three important prerequisites before you start:

- The relevant ODBC driver must be installed on your machine.
- You must have local administrator rights to set up the ODBC connection.
- You must have connection details for the target database, including a username and password to authenticate.

**How to set up an ODBC connection in Microsoft Windows 7**

1. Open **Control Panel** and then **Administrative Tools**. If your control panel is organized by theme, you'll have to go into **System and Security** and then into **Administrative Tools**.

2. In **Administrative Tools**, open **Data Sources (ODBC)** and click on the **System DSN** tab.
3. Click on the Add button to configure the connector, scrolling down the list of drivers until you find the one you need. For this example, we're interested in the Oracle driver.

![Create New Data Source](image)

4. Click on Finish to get to the business end of the configuration.

![Microsoft ODBC for Oracle Setup](image)

5. **Data Source Name** is entirely arbitrary; it is the reference you will use in Spotfire. The **Server** configuration includes a hostname, which is simply the name of the server hosting the database; a port, usually 1521; and a DB service name, which you can get from the database administrator. Click on OK to finish and return to the **System DSN** tab, where you will see your entry. Make sure you click on OK to exit the ODBC administration tool.

Now you are ready to use this ODBC connection to retrieve data into Spotfire.
Using an ODBC connection in Spotfire

Use the File menu or the Add Data Table icon to navigate to the add data table dialog just as you did to add data from a file, except this time you are going to select Database rather than File.

1. Click on Database to get the Open Database dialog. We're interested in Odbc Data Provider, so select it and click on the Configure button.
2. Select the connection name you configured in the ODBC administrator. You'll notice that a number of connectors are defined in the next screenshot, including one called SpotfireDev. You will need to create one of your own for a database to which you have access.

3. Click on OK to get back to the Open Database window; notice that Spotfire has created a connection string. Click on OK to open the database. The database will eventually open; it may take a little time if your network connection is slow or the database is complex. You will see a hierarchical organization of high-level containers, in each of which you should see data tables and their constituent columns. It's not unlike navigating a folder system on your computer.
4. You will need to have some knowledge of the database and some idea of the data you are looking for, but Spotfire will allow you to navigate the database objects and select whole or partial tables to import. Spotfire will also generate the SQL automatically, but you can customize and fine-tune the SQL, adding a where clause, for example.

5. If you click on + next to one of the main objects, such as SPOTFIRE in the example, the entry will expand to list all the tables in that schema. You can then scroll down to a table of interest and expand further to view the individual columns in the table.

![Specify Tables and Columns](image)

6. When you are happy with the selections, give the import a name. This will be the name of the table in Spotfire. The data will load just as the files did, except the structure of the data (column headers and data types) will be determined by the source database.

**Downloading the example code**
You can download the example code files for all Packt books you have purchased from your account at [http://www.packtpub.com](http://www.packtpub.com). If you purchased this book elsewhere, you can visit [http://www.packtpub.com/support](http://www.packtpub.com/support) and register to have the files e-mailed directly to you. You can also download code from [http://www.insidespotfire.com](http://www.insidespotfire.com).
Key data concept – data queries

Our final data concept in this chapter is the idea of a data query, which is usually achieved using some flavor of SQL, a standard language to interact with databases. Although SQL varies a little from database to database, the basic syntax is fairly consistent. If you don't already have experience using SQL and you want to develop your data analysis skills, you will have to learn more about this key analytics tool.

When you work with text files and spreadsheets, the only practical way you can manipulate large amounts of data is at the column level, removing entire columns that you don't want. What if you want to manipulate the rows in a large dataset? In a spreadsheet, you could of course, use filters to remove rows.

Overall, however, manipulating data in this way is more tedious and potentially more error prone than using a single SQL statement. What if you want to involve more than one table in your "filter"? You can also take advantage of the enormous power of databases to handle large amounts of data and process complex queries.

Anatomy of a SQL statement

A basic SQL statement has just three elements, highlighted in the following code in all caps:

```
SELECT column_name_1, column_name_2, ..., column_name_n
FROM schema_name.data_table_name
WHERE condition
```

You need to provide a `schema_name`. Remember when we connected to a database, we saw a set of logical containers into which the tables were organized. These are examples of schemas. The terminology `schema` may mean slightly different things in different databases, but most organize the tables into a set of logical containers, and you need to prefix table names in your query with the container name using a dot to separate the two elements.

Often, a database administrator will create what is known as a `view`. These are pre-written, often quite complex, queries spanning multiple tables that define a particular dataset. As an end user, you can use them just like tables.

If you have a complex data requirement that you feel is beyond your SQL skills, talk to a database administrator or developer and get them to create a view for you. You can then select from the view as though it were a single table.
The `SELECT` and `FROM` clauses are mandatory, but the `WHERE` condition is optional and is just a logical expression to limit the data that is returned. For example:

```
WHERE column_name_1 = 'some text' AND column_name_2 > 4
```

If you want to include more than one table in your query, you will have to use what is known as a `join`.

```
SELECT a.column_name_1, b.column_name_2, …, etc
FROM schema_name.data_table_name_1 a
JOIN schema_name.data_table_name_2 b on a.column_name_x = b.column_name_y
WHERE condition
```

Note how the aliases `a` and `b` have been used for the two tables. Aliases can be anything you like, but are usually short and make the statement easier to write and read.

The `JOIN` statement ensures that you only return rows where the join condition is true, as well as any other condition you defined. There are other types of joins. The following link of the w3schools website provides a helpful tutorial: [http://www.w3schools.com/sql/sql_join.asp](http://www.w3schools.com/sql/sql_join.asp)

### Summary

You have a come a long way in this chapter, and you should be pleased with the progress you have made. By now you should know how to add data to Spotfire from text, spreadsheet, and data files; link data in Spotfire; and connect Spotfire to a database.

We also covered Spotfire concepts like the inbuilt filters, data table properties, and table relations and column matches in Spotfire.

Finally, you picked up a lot of essential data concepts like the basic row/column structure of a data table, different date types, data relationships, how to use ODBC, and data queries using SQL.

I hope this chapter has not been too dull. It is very important that you understand data sources, how to import data into Spotfire, and how to manipulate data. Even the most ingenious visualization or collection of visualizations is only as good as the underlying data. If you get the data part wrong or you can't get the data you want, you are wasting your time, at best, and grossly misleading your analytics consumers, at worst.

This chapter is just a basic foundation in data analysis, but it is sufficient and will serve you well to explore the more visually compelling aspects of Spotfire. Let's go straight to Chapter 2, Visualize This! and start visualizing!
Where to buy this book

You can buy TIBCO Spotfire – A Comprehensive Primer from the Packt Publishing website.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.