

Data Preparation for Qlik Sense Desktop using Pentaho Kettle

Stephen Redmond, CTO and Qlik Luminary





TABLE OF CONTENTS

GETTING SET

[Introducing Kettle](#)

[Introducing Qlik Sense Desktop](#)

[Preparing the software prerequisites](#)

[Introducing the pagila database](#)

[Summary](#)

GETTING STARTED WITH KETTLE

[Running Kettle for the first time](#)

[Creating our first transformation](#)

[Summary](#)

BLENDING DATA

[Using the Stream lookup](#)

[Extending the model](#)

[Combining transformations](#)

[Using Joins](#)

[Summary](#)



MODIFYING DATA

[Using Calculator to derive calendar fields](#)

[Examining the formula tool](#)

[Creating new data](#)

[Summary](#)

DISCOVERING WITH QLIK SENSE DESKTOP

[Creating a new app in Qlik Sense Desktop](#)

[Loading data from files](#)

[Creating our first analysis](#)

[Using images and text](#)

[Telling stories](#)

[Sharing apps with other users](#)

[Summary](#)

APPENDIX

[Installing the pagila database in PostgreSQL](#)

[Connecting to the pagila database from Kettle](#)



CHAPTER ONE

GETTING SET



Getting set

In this chapter we are going to have a look at the software that we will use throughout the rest of this book. We will have a brief introduction to each of the pieces and run through how to install each of them on your Windows PC.

These are the topics we'll be covering in this chapter:

- Introducing Kettle
- Introducing Qlik Sense Desktop
- Preparing the software prerequisites
- Introducing the pagila database



Introducing Kettle

Pentaho Data Integration – Kettle – is a tool that allows a user to build data extraction routines, transform that data, and load the finished data into different format for further use – for example in Qlik Sense.

Understanding why using Kettle makes sense for a business

There are several reasons why using Kettle makes sense for a business:

- Low cost of entry – the Community Edition of Kettle is free to download and use.
- Quick to learn – the drag/drop intuitive interface allows users to create useful content very quickly
- Solid technology – although “open source” may not always inspire confidence, Pentaho also has a commercial operation. The technology has been well tested and implemented widely. At the time of writing, Hitachi have announced that they will acquire Pentaho.
- Ability to scale – business can start small with Community Edition and then scale upwards with the Enterprise Edition onto multiple servers, on premise or in the cloud.
- Excellent integrations – Kettle comes out-of-box with integrations to all of the major database systems, including Oracle, Microsoft SQL Server, MySQL, DB2, Vertica, PostgreSQL and Hadoop. New connections can be created with an open API.

Introducing Qlik Sense Desktop

Building on 20 years of experience in Data Discovery with QlikView, Qlik have released a new generation product called Qlik Sense that allows users to consume data from multiple data sources and create their own interactive, visual dashboards. It is very much targeted at the Self-Service Data Discovery space.

Qlik Sense Desktop is a Windows application that allows users to quickly create new dashboards for personal discovery or for sharing with colleagues.



Qlik Sense can connect to many data sources and Qlik Sense Desktop excels at consuming desktop data sources without the user having to learn any data scripting. Using an easy-to-use data preparation tool such as Kettle to create data files to be consumed by Qlik Sense makes a great self-service partnership for business users.

Understanding why using Qlik Sense Desktop makes sense for business

There are several reasons why using Qlik Sense makes sense for a business, and they are very similar to the reasons for using Kettle:

- Low cost of entry – Qlik Sense Desktop is free to download and use.
- Quick to learn – files can be dragged into Qlik Sense Desktop to allow quick access to their data. The simple interface in Qlik Sense allows users to quickly create visual content to explore and discover.
- New generation of a great technology – QlikView has been around since 1993 and has been the market leader in in-memory BI and Data Discovery. Qlik Sense is the next generation of the QlikView in-memory engine using a modern, touch aware user interface.
- Ability to scale – business can start small with the free Qlik Sense Desktop and then move their applications to Qlik Sense Server, scaling across multiple servers, on premise or in the cloud.
- Multi-data source integration – Qlik Sense will connect out-of-box to any OLEDB or ODBC data sources. Qlik also has connectors for SAP, Google Big Query, Cloudera, Salesforce and many others.

Preparing the software prerequisites

To follow the steps in this book, you are going to need two main applications:

- Pentaho Data Integration (Kettle)
- Qlik Sense Desktop

The data used for the examples in this book is based off the PostgreSQL demonstration database, pagila ([pagila-0.10.1.zip](http://pgfoundry.org/frs/?group_id=1000150&release_id=998)), which is available from: http://pgfoundry.org/frs/?group_id=1000150&release_id=998

Instructions for installing PostgreSQL and the pagila database will also be provided here for those who have an interest.

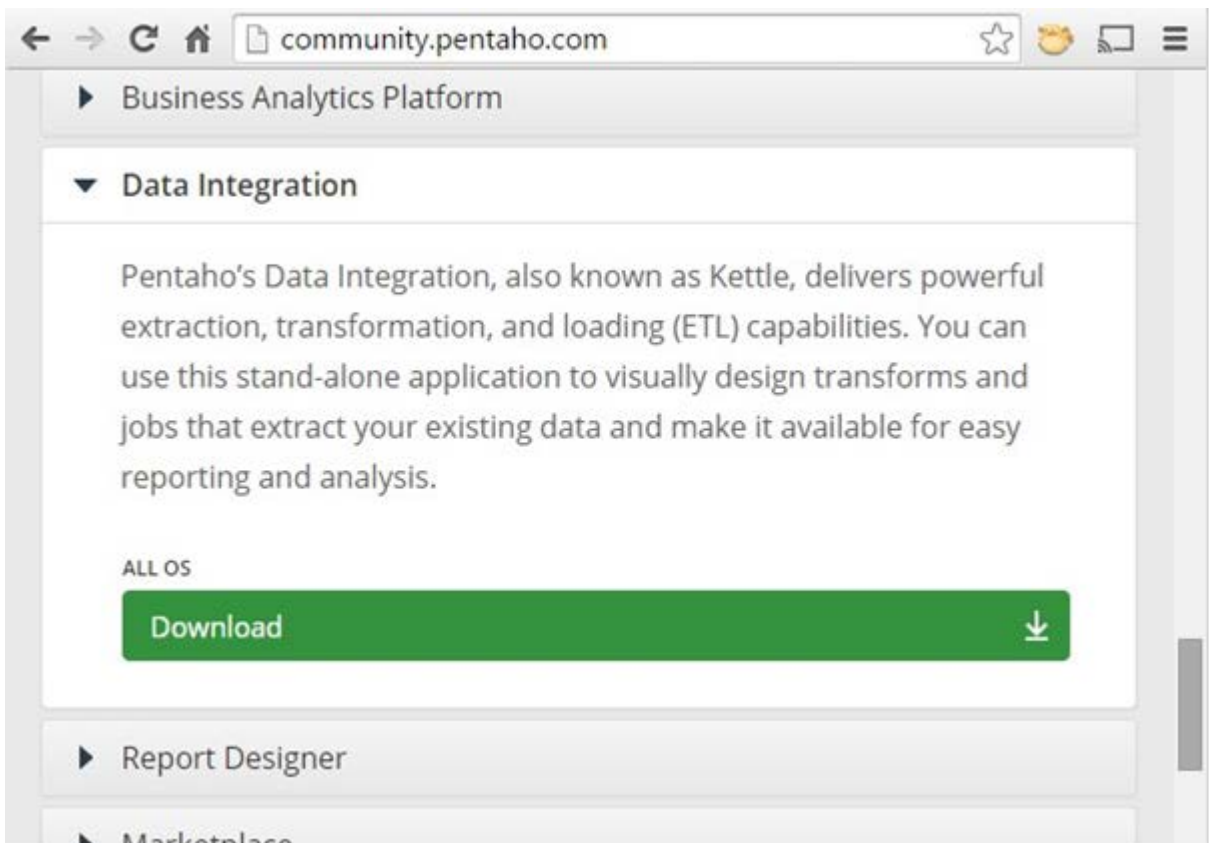


Installing Kettle

These instructions tell us how to download and install the Pentaho Data Integration (Kettle) Community Edition.

Downloading Kettle

To download Pentaho Data Integration (Kettle) Community Edition, first open the Pentaho Community site, <http://community.pentaho.com/>, in your web browser. Scroll down to the download area and open the **Data Integration** section.



Click on the Download link and save the zip file to a local folder.



Unzipping Kettle

Locate the zip file that you downloaded above (for example, [pdi-ce-5.3.0.0-213.zip](#)). Right-click on it and select **Extract All...** from the menu.

Enter an appropriate folder name, for example, C: \Pentaho, and click **Next**. It will create a sub-folder called `data-integration`.

Open the `data-integration` folder and locate the file `spoon.bat` – double-click it to run it. You should see a **Spoon tips...** dialog, like in this screenshot:



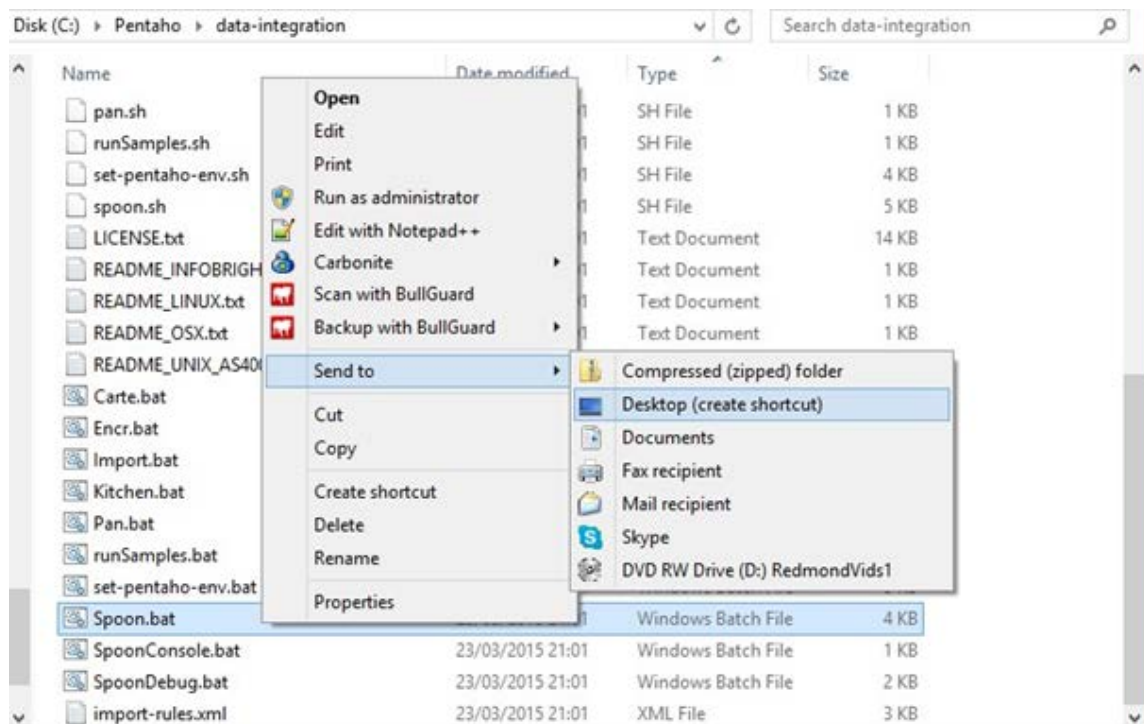
Close the tip and the **Spoon** main window will appear. You can close the Spoon window for now.



Creating a desktop shortcut

Because it can be difficult to constantly remember where the correct file is, it can be useful to create a desktop shortcut to run Spoon.

Again, locate the spoon.bat file in the data-integration folder. Right-click on it and select **Send to** and then **Desktop (create shortcut)**.



Locate the new shortcut on the desktop (it should be called **Spoon.bat – Shortcut**). Right-click on the file and select Rename from the menu. Rename it to **Kettle Spoon**.

Installing Qlik Sense Desktop

These instructions will show us how to download and install Qlik Sense Desktop.



Downloading Qlik Sense

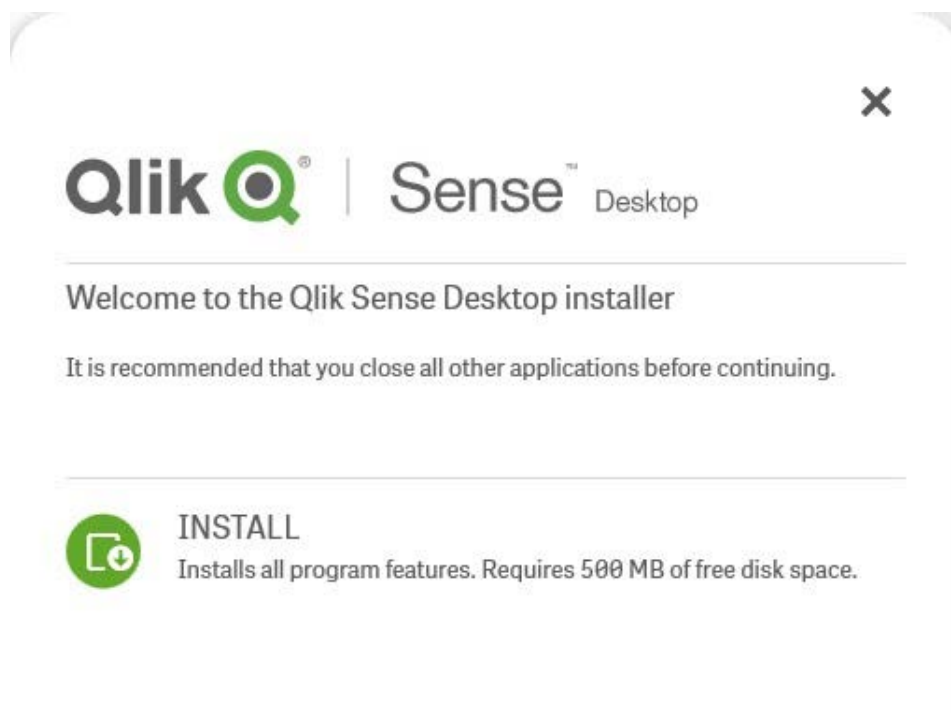
Go to the Qlik download page at

<http://www.qlik.com/us/explore/products/free-download> and click on **Get Qlik Sense Desktop**. Fill in the registration information and click on **Download Now**.

*Note that if you have a Qlik single sign-on that entitles you to download software (Customer or Partner) then you should instead, at the main download page, select the **Get QlikView Personal Edition** option as this gives access to the application that allows you to download all the Qlik software that you have access to.*

Running the installer

Once the application has been downloaded, double-click on the file (Qlik_Sense_Desktop_setup.exe) to launch.



Click the **INSTALL** button, accept the license agreement on the next screen and then click **Next**. Leave the **Create desktop shortcuts** checkbox on and click **Install**.



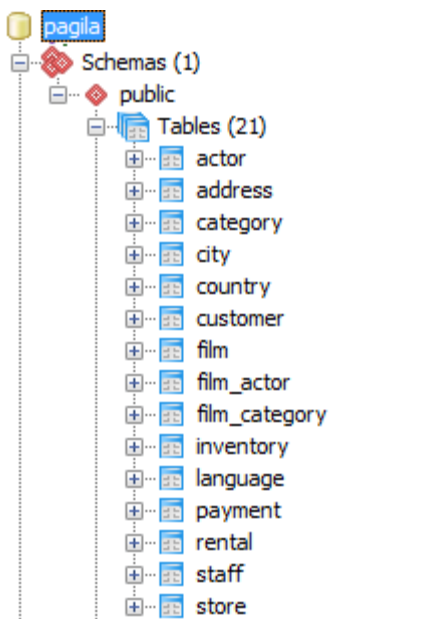
Once the installation has finished, you can click **Finish**.

To test the installation, launch **Qlik Sense Desktop** from the desktop shortcut. The **Qlik Sense Desktop Hub** should appear.

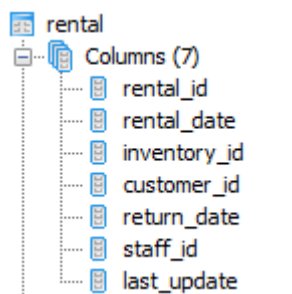
Introducing the pagila database

The pagila database is a port of the Sakila demo database used in MySQL, which is based on an original Dell project about an online DVD store.

The database has fifteen tables that record the details of DVD rentals.



The main table is, of course, the **rental** table.





We can see that the **rental** table has links to **inventory** (which tells us what **film** and what **store** this rental refers to), **customer** (which has links to **address**, **city** and **country**) and **staff** (which also has links to **address**, **city** and **country**). The **payments** table records payment information for rentals. The **film** table has links to **actor** (via **film_actor**), **film_category** and **language**.

The **address** table is interesting because it links to **customer**, **store** and **staff**. However, it is unlikely that we will need to analyse it as a single table and more likely that we will want to analyse as customer address, store address and staff address.

If you are interested in installing PostgreSQL and the pagila database, full instructions are included in the Appendix.

Summary

In this chapter we have introduced the technologies that we will use to enable self-service data discovery and we have learned how to download and install them. We have also had an introduction to the data that we are going to use to learn how to use the tools.

In the next chapter we will start to learn how to use Kettle to load and transform data and then load it into files that we can use in Qlik Sense.



CHAPTER TWO

GETTING STARTED WITH KETTLE



Getting started with Kettle

In this chapter we will begin to learn how to prepare data in Pentaho Kettle. These are the topics we'll be covering in this chapter:

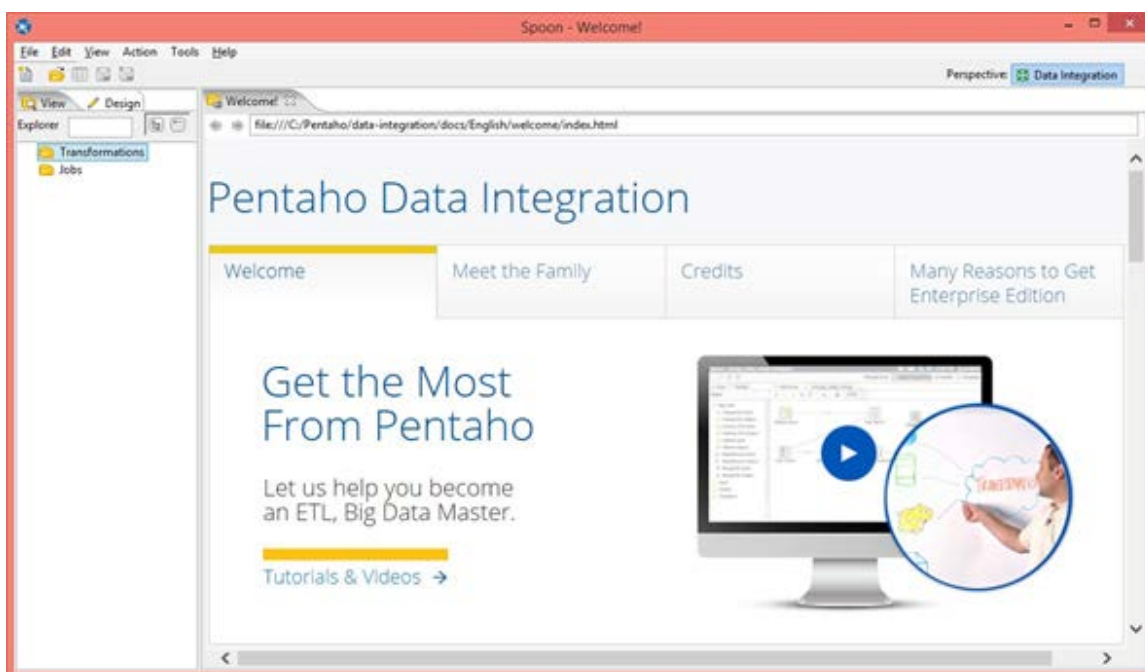
- Running Kettle for the first time
- Creating our first transformation



Running Kettle for the first time

Let's use the desktop shortcut that we created in *Chapter 1, Installing Kettle*. Locate the shortcut and double-click on it.

Note that a command prompt box will appear and disappear and there may be a few seconds where it appears that nothing is happening. Be patient and the splash screen will appear. After a few more seconds, the main Spoon window appears.

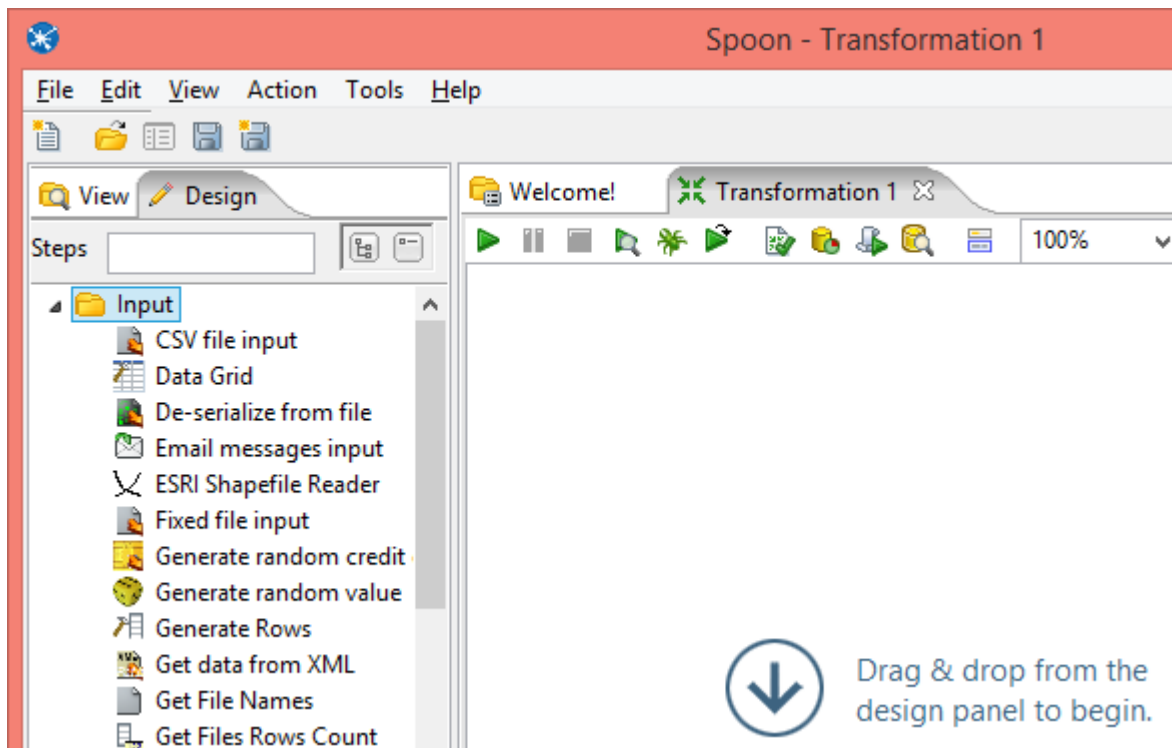


Let's create a new Transformation to get started. Click the **New File** icon in the toolbar (it is the first icon in the toolbar) and select **Transformation** from the menu. You can also create a new transformation from the **File** menu or by using the keyboard shortcut, **Ctrl+N**.



Getting to know the Spoon development environment

There are three main areas of the development environment – the left panels, the design area and the **Execution Results** panel.



Left panels

There are two tabs in this panel – **View** and **Design**.

The **View** panel lists out all the Transformations and Jobs that are open and gives access to various properties. Most basic users will not need to use this tab too much.

The **Design** panel lists all the various tools that we can use in our transformations or jobs.



Design area

The design area is where we will create our transformations by dragging tools from the **Design** panel. We will start to get to know this area very well very soon.

Execution Results panel

This panel, which will sit below the Design area, will only be visible once we have executed or preview executed our transformation. It gives useful metrics on performance as well as giving us the opportunity to preview data.

Creating our first transformation

We will use the new transformation that we created in the previous section.

*To recap creating a transformation. Click the **New File** icon in the toolbar (it is the first icon in the toolbar) and select **Transformation** from the menu. You can also create a new transformation from the **File** menu or by using the keyboard shortcut, **Ctrl+N**.*

Adding an input

Input steps are the primary way of getting data into the transformation. Open out the **Input** folder under the **Design** tab on the left panel:



- Input
 - CSV file input
 - Data Grid
 - De-serialize from file
 - Email messages input
 - ESRI Shapefile Reader
 - Fixed file input
 - Generate random credit card numbers
 - Generate random value
 - Generate Rows
 - Get data from XML
 - Get File Names
 - Get Files Rows Count
 - Get repository names
 - Get SubFolder names
 - Get System Info
 - Get table names
 - Google Analytics
 - GZIP CSV Input
 - HL7 Input
 - Json Input
 - LDAP Input
 - LDIF Input
 - Load file content in memory
 - Microsoft Access Input
 - Microsoft Excel Input
 - Mondrian Input
 - OLAP Input
 - Property Input
 - RSS Input
 - S3 CSV Input
 - Salesforce Input
 - SAP Input
 - SAS Input
 - Table input
 - Text file input
 - XBase input
 - XML Input Stream (StAX)
 - Yaml Input

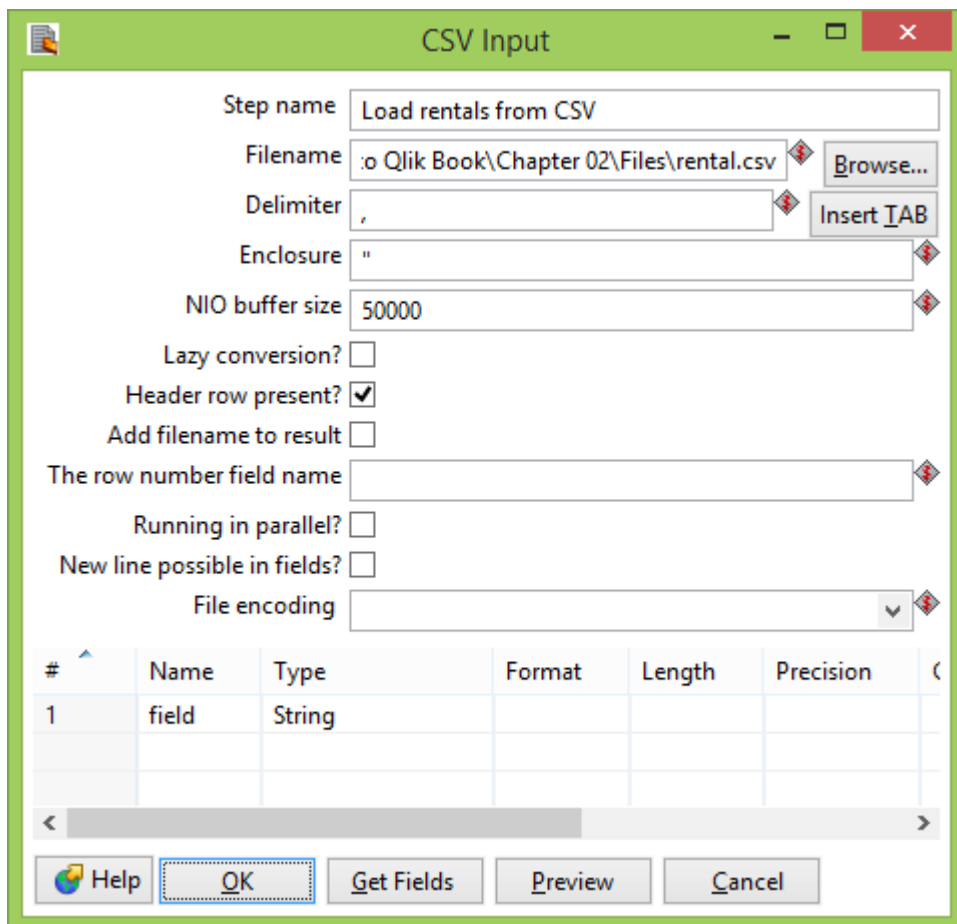
We can see that there are a large number of inputs ranging from CSV files to Salesforce. These are not the only inputs available. For example, there are others listed under **Big Data**, **Palo** and **Open ERP**.

Note the search box in the Design tab. If we type text in here it will list all the tools that match that text – it is a great way to find a tool quickly.

We want to read data from one of our CSV files so let's drag a CSV file input onto our transformation.



When we hover over a step in the transformation we will see a tool menu appear. The one that we will use the most is the pencil icon to edit the step. We can also edit the step by double-clicking. Let's double-click this step that we have just added – the **CSV Input** dialog will open



Step name: Load rentals from CSV

Filename: .o Qlik Book\Chapter 02\Files\rental.csv Browse...

Delimiter: , Insert TAB

Enclosure: "

NIO buffer size: 50000

Lazy conversion?

Header row present?

Add filename to result

The row number field name:

Running in parallel?

New line possible in fields?

File encoding:

#	Name	Type	Format	Length	Precision
1	field	String			

Buttons: Help, OK, Get Fields, Preview, Cancel



Enter an appropriate name for step and then click the **Browse...** button. Select the [rental.csv](#) file from the downloaded files folder.

Make sure that the delimiter is set to comma. Also, deselect the **Lazy conversion** checkbox.

Lazy conversion is a useful option if we are loading data from a file and doing minimal processing before pushing the information elsewhere. With the checkbox on, Kettle will not try and convert data to dates or numbers.

Click the **Get Fields** button. We are prompted to **Enter the number of lines to sample** and the default is **100**. This means that Kettle will scan down the first 100 lines to establish what type of data we have – strings, numbers, dates, etc. After we click OK we will get a report of the discovery. Close this report and we should see the field list:

#	Name	Type	Format	Length	Pre
1	rental_id	Integer	#	15	0
2	rental_date	Date	yyyy/MM/dd HH:mm:ss.SSS		
3	inventory_id	Integer	#	15	0
4	customer_id	Integer	#	15	0
5	return_date	Date	yyyy/MM/dd HH:mm:ss.SSS		
6	staff_id	Integer	#	15	0
7	last_update	Date	yyyy/MM/dd HH:mm:ss.SSS		

Click **OK**.



Saving the transformation

Before we go any further, we should save our work. Note that transformations cannot be executed without saving and you will be prompted to do so.

Select **Save** from the **File** menu or click the Save button on the toolbar. Let's call this transformation `My First Transformation`.

Adding a Dummy step

We could actually run this transformation now but nothing much would happen. To be able to even preview the output of the load step, we need to output the results of the step. So, we need to have somewhere to output those results to. Later, we will add real steps to do something with that output, but for now we are going to add a Dummy step.

A Dummy step will just give us somewhere to send the output. It doesn't do anything with it, but it will mean that we can test our transformation without having to write out any files.

Drag a **Dummy (do nothing)** tool from under the **Flow** folder of the **Design** tab onto the transformation.

Adding our first hop

A hop tells the connection between different steps. For one step the hop will contain the output, and for the other step it will be the input. Some steps may have more than one type of output. For example, as well as the main output, some steps may have the option to send errors as output. We are generally only interested in the main output.

The easiest way to add a hop is the click on one step and then, with the *Shift* key pressed, hold the mouse button down and drag onto the second step.





Kettle will prompt us for whether we want to use the **Main output of step** or **Error handling of step**
– we want to choose the main output.

Running the transformation

Before running the transformation, we should save what we have done by clicking the Save icon on the main toolbar. If we don't we will be prompted anyway before the transformation is launched.

From the **Action** menu, select **Run** or click the **Run this transformation or job** button in the transformation's toolbar. The **Execute a transformation** dialog will open:

Execute a transformation

Local, remote or clustered execution

Local execution Execute remotely Execute clustered

Remote host

Pass export to remote server

Post transformation
 Prepare execution
 Start execution
 Show transformations

Details

Enable safe mode
 Gather performance metrics
 Clear the log before execution

Log level: Basic logging

Replay date (yyyy/MM/dd HH:mm:ss)

Parameters

#	Parameter	Value	Default value
1			

Variables

#	Variable	Value
1	Internal.Job.Filename.Directory	Parent Job File Directory
2	Internal.Job.Filename.Name	Parent Job Filename
3	Internal.Job.Name	Parent Job Name
4	Internal.Job.Repository.Directory	Parent Job Repository D

Arguments

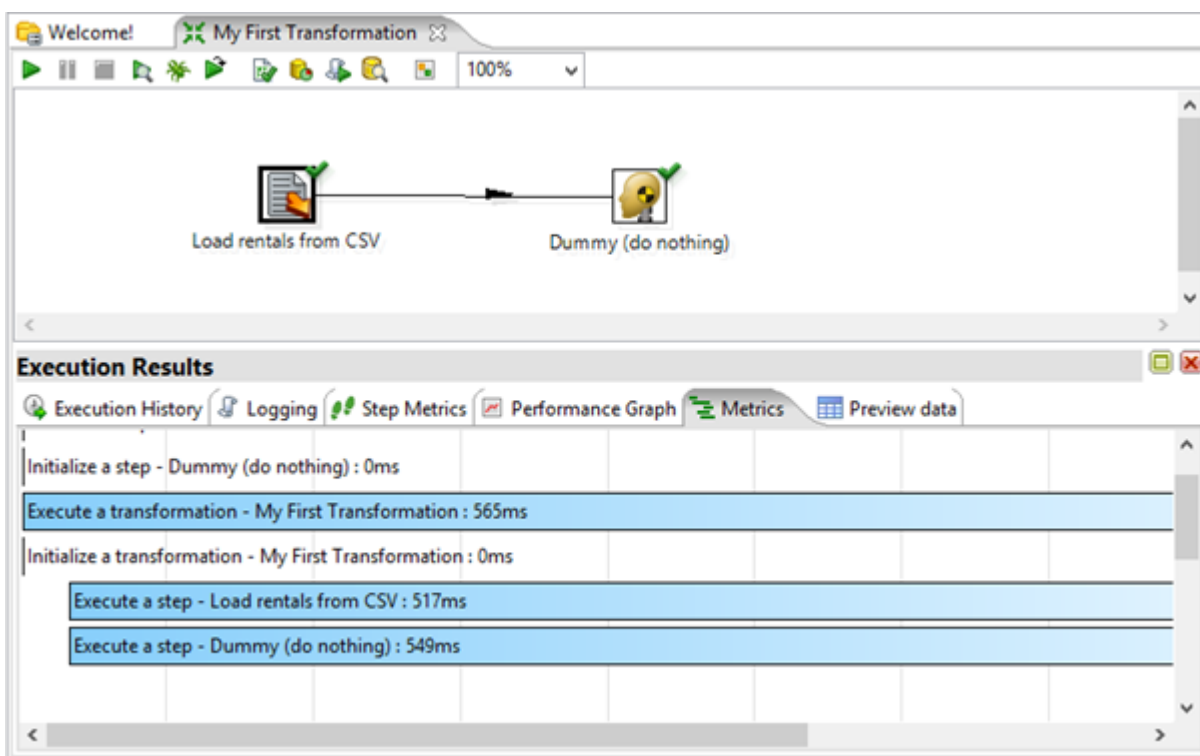
#	Argument	Value
1		

Launch Cancel



This dialog allows us to set some parameters that we don't really need to worry about at this stage. We just need to click the **Launch** button.

If all goes well, we should see tick marks appear beside each step. We should also now see the **Execution Results** panel and be able to see execution **Metrics** and **Preview Data**.



Outputting data

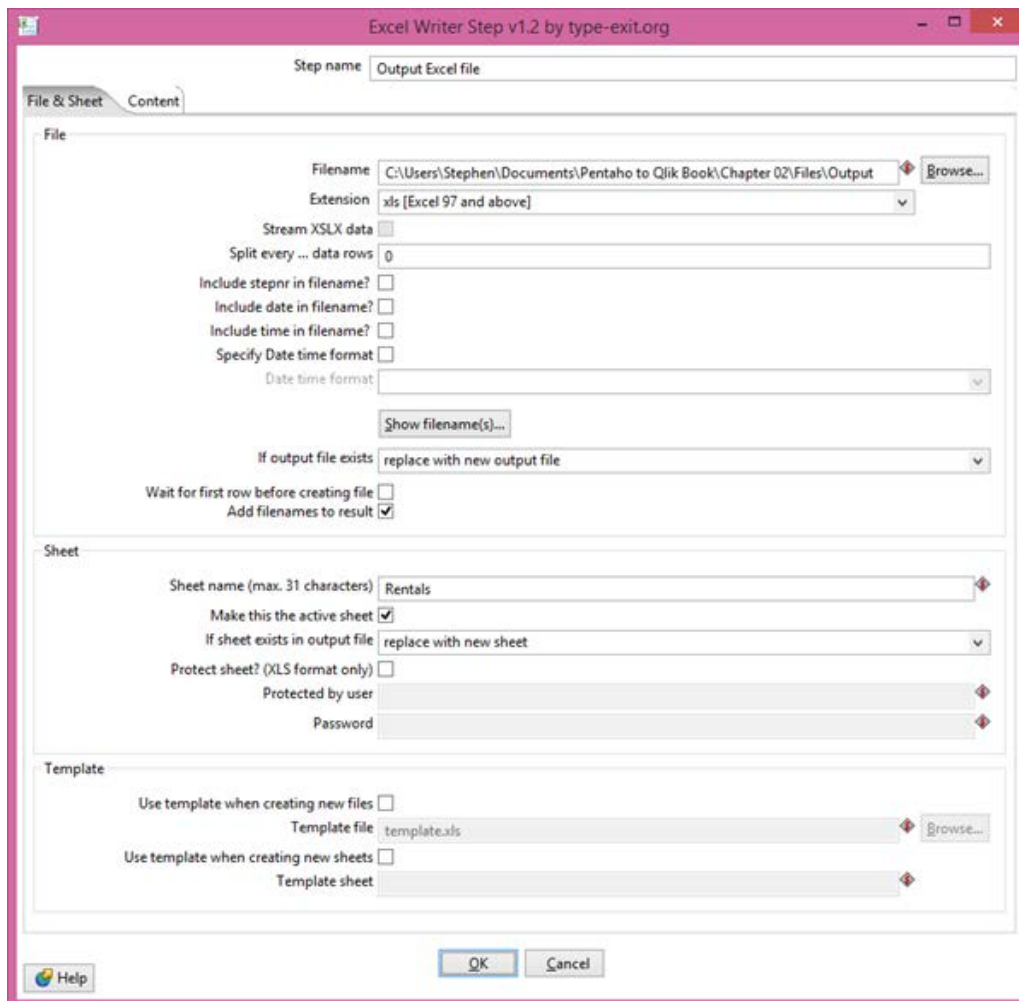
Using the Dummy step is great for testing but, at the end of the day, we really want to be able to write data out to a file so that it can be consumed in Qlik Sense. We can go ahead and delete the Dummy step now by clicking on it to select it (the outline will be thicker) and clicking the *Delete* key.

Kettle can output data in many different formats. For use with Qlik Sense, either Excel, Text or XML are all suitable outputs.



For this example, we will write our data to an xls file. Drag a **Microsoft Excel writer** tool from the **Output** folder in the **Design** panel onto the transformation. As before, create the hop by clicking and dragging from the CSV load step to the new Excel step while holding the *Shift* key. Again, we want the main output. Double-click the Excel step to configure it.

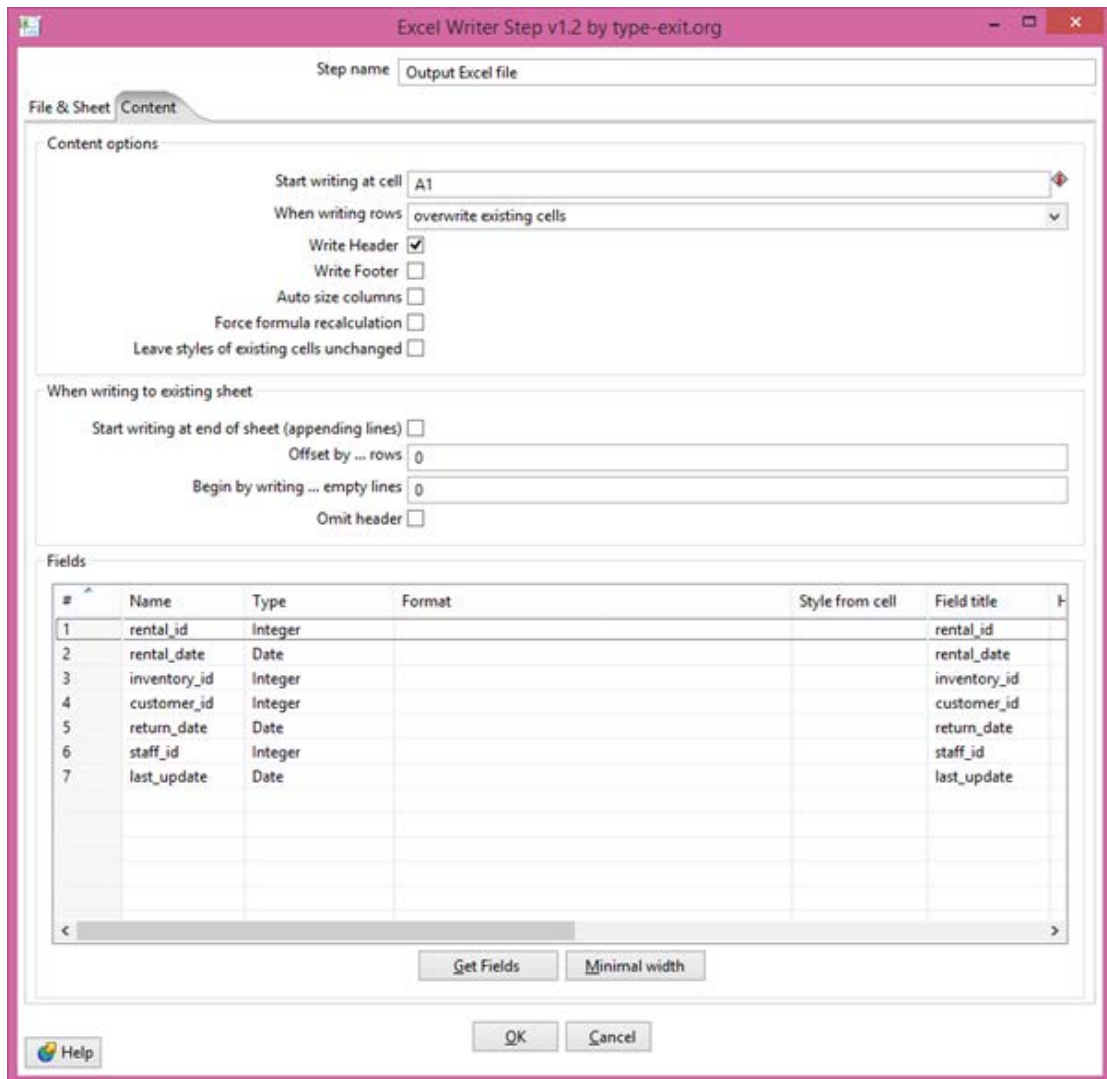
Note that as at time of writing there is an issue with Java 8 that means the Excel writer will successfully create .xlsx files but then throw an error. Therefore we are going to use .xls files in this example. This error should be fixed in Java 9. There is no problem with Java 7.





Note now that there are two tabs in this dialog – **File & Sheet** and **Content**.

In the first tab, we will enter a path and name for our file (don't add the .xls!). In the **Extension** drop-down, we will select **xls [Excel 97 and above]**. In the **Sheet name** box, we will enter **Rentals**. Click onto the **Content** tab:





Click the Get Fields button to retrieve the list of fields from the output of the first step. This is the list of fields that we will write to the Excel file. We could, if we wanted, select any row and delete if we didn't want to export it.

Click **OK** to save the step. Save the transformation and run it as before.

If all is well then we should have an Excel file in our output folder that contains the rental information.

Summary

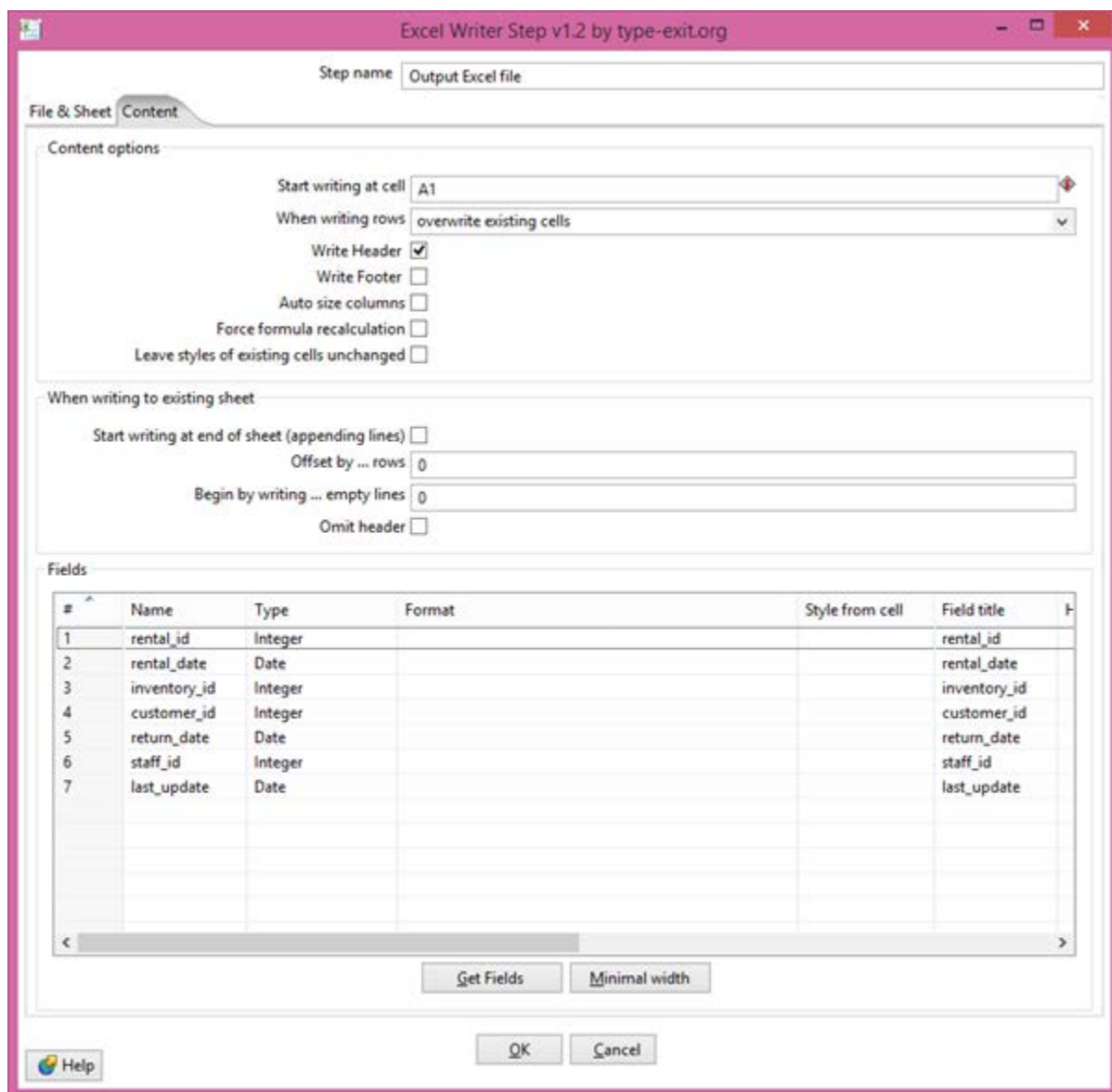
In this chapter we have learned about getting started with Kettle. We know where most things are on the layout and we know how to create new transformations. We have created a transformation that loads data from a CSV file and writes it to an Excel file that we can use with Qlik Sense.

In the next chapter we will bring in additional data and see how we can merge it with our rental data to create a single output.



Note now that there are two tabs in this dialog – **File & Sheet** and **Content**.

In the first tab, we will enter a path and name for our file (don't add the .xls!). In the **Extension** drop-down, we will select **xls [Excel 97 and above]**. In the **Sheet name** box, we will enter **Rentals**. Click onto the **Content** tab:





Click the Get Fields button to retrieve the list of fields from the output of the first step. This is the list of fields that we will write to the Excel file. We could, if we wanted, select any row and delete if we didn't want to export it.

Click **OK** to save the step. Save the transformation and run it as before.

If all is well then we should have an Excel file in our output folder that contains the rental information.

Summary

In this chapter we have learned about getting started with Kettle. We know where most things are on the layout and we know how to create new transformations. We have created a transformation that loads data from a CSV file and writes it to an Excel file that we can use with Qlik Sense.

In the next chapter we will bring in additional data and see how we can merge it with our rental data to create a single output.



CHAPTER THREE

BLENDING DATA



Blending Data

One of the key reasons to use a data preparation tool, like Kettle, is so that we can combine lots of different data sources together to produce our final output. In this chapter we will examine some different ways of combining data, starting with one of the most powerful, the Stream lookup.

These are the topics we'll be covering in this chapter:

- Using the Stream lookup
- Extending the model
- Combining transformations
- Using Joins



Using the Stream lookup

One of the easiest ways to blend data in Kettle is to use the Stream lookup. It is also a very powerful way of blending data because it can avoid some of the pitfalls of other joining methods and has some features that are unique to this method and are very useful.

In this section, we will create a single table containing Customer information. This will include data from the **customer** table, **address** table, **city** table and **country** table.

There is also a Database lookup option that can be used if the source is a database table.

Introducing the Stream lookup

The Stream lookup will take inputs from two preceding steps. One of those steps is defined as the lookup table and the other step will have a field containing values that we will look up in that lookup table. It has some similarities to the way VLOOKUP works in Excel.

If there are more than one row in the lookup table that would match the search, then only one of them will be returned – the last one in the table. This avoids potential issues with creating duplicates using other join methods.

Looking at a simple use of Stream lookup

Let's create a brand new transformation in Kettle. As before, the first thing to do is save it. Let's call it `Customer Transformation`.

Add a new CSV input to the transformation. Edit the properties of this input and set the file to load to be `country.csv` from the data files. As before, click the Get Fields button to retrieve the list of fields and their types from the data.



CSV Input

Step name: Load Country from CSV

Filename: Qlik Book\Chapter 02\Files\country.csv

Delimiter: ,

Enclosure: "

NIO buffer size: 50000

Lazy conversion?

Header row present?

Add filename to result

The row number field name:

Running in parallel?

New line possible in fields?

File encoding:

#	Name	Type	Format	Len
1	country_id	Integer	#	15
2	country	String		37
3	last_update	Date	yyyy/MM/dd HH:mm:ss.SSS	

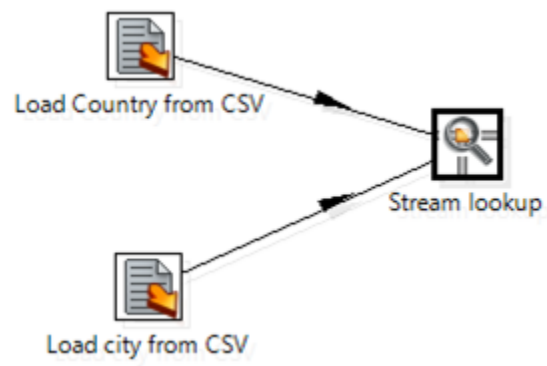
Now, we won't need the **last_update** field so click on the number 3 to select that row and press the *Delete* button on the keyboard to remove it. Click **OK**.

*To avoid potential conversion issues later down the track, it is a good idea to uncheck the **Lazy conversion** checkbox. There is some overhead to this, but not an issue on smaller datasets.*

Add a second CSV input to the transformation. In this case, configure it to load the city.csv file. We can also remove the **last_update** field from this load.



From the **Design** tab, under the **Lookup** folder, drag a **Stream lookup** tool onto the transformation. Using the same method as before (hold the *Shift* key while clicking and dragging), add a hop from each CSV input tool into the Stream lookup, selecting the main output of the step in each case:



Double-click the Stream lookup to edit the properties:

The screenshot shows the 'Stream Value Lookup' dialog box with the following configuration:

- Step name: Lookup country
- Lookup step: Load Country from CSV
- The key(s) to look up the value(s):

#	Field	LookupField
1	country_id	country_id
- Specify the fields to retrieve:

#	Field	New name	Default	Type
1	country		**Missing**	String
- Preserve memory (costs CPU)
- Key and value are exactly one integer
- Use sorted list (i.s.o. hashtable)

Buttons at the bottom: Help, OK, Cancel, Get Fields, Get lookup fields.



The first thing to set here is to specify which of the two steps that we linked into the lookup is the Lookup step. In this case, it is the country step.

We then need to pick the **Field** that will have the values we want to use to search with. In this case, it is **country_id**. Then we specify the **LookupField**, which is the field in the country step that we are going to compare – this is also **country_id**.

Finally, we specify the field that will be retrieved by the lookup – in this case, the **country** field. We can also specify a default value that should be used in case the lookup fails. We will specify a value of ****Missing**** - which will be easily seen in the data to flag that issue.

The lookup might fail if we looked for an id that doesn't exist in the country table. This feature of being able to specify a default value is really very important.

Add a **Dummy** tool to the transformation and connect a hop from the Stream lookup to the Dummy. Save and launch the transformation. Once the transformation is complete, click on the Dummy step and you can preview the results in the **Execution Results** panel:

#	city_id	city	country_id	country
591	591	Yuzhou	23	China
592	592	Zalantun	23	China
593	593	Zanzibar	93	Tanzania
594	594	Zaoyang	23	China
595	595	Zapopan	60	Mexico
596	596	Zaria	69	Nigeria
597	597	Zeleznogorsk	80	Russian Federation
598	598	Zhezqazghan	51	Kazakstan
599	599	Zhoushan	23	China
600	600	Ziguinchor	83	Senegal



Extending the model

In the previous section we used the Stream lookup to do a simple data blend by taking one field from the country table and merging it in with the data from the city table. In this section, we will bring in the address information and blend it with our combined city and country data.

Adding the address

Right-click on the hop between the country Stream lookup and the Dummy and select **Delete hop** from the menu. We will keep the Dummy tool on the transformation for the moment because we will use it later.

Add a new CSV input tool to the transformation. Configure it to load the address.csv file. Again, we don't need the **last_update** field.

Merging address, city and country

Add a new Stream lookup onto the transformation. Connect hops to this new Stream lookup from both the address CSV input and the original stream lookup (that blended city and country).

Double-click the new Stream lookup to configure. The **Lookup step** should be the previous Stream lookup. The lookup keys will both be `city_id`.

We are going to do something slightly different here in that we are going to retrieve two fields from the lookup instead of one. We will retrieve both the city and country fields:



Stream Value Lookup

Step name: Lookup city and country

Lookup step: Lookup country

The key(s) to look up the value(s):

#	Field	LookupField
1	city_id	city_id

Specify the fields to retrieve :

#	Field	New name	Default	Type
1	city		**missing**	String
2	country		**missing**	String

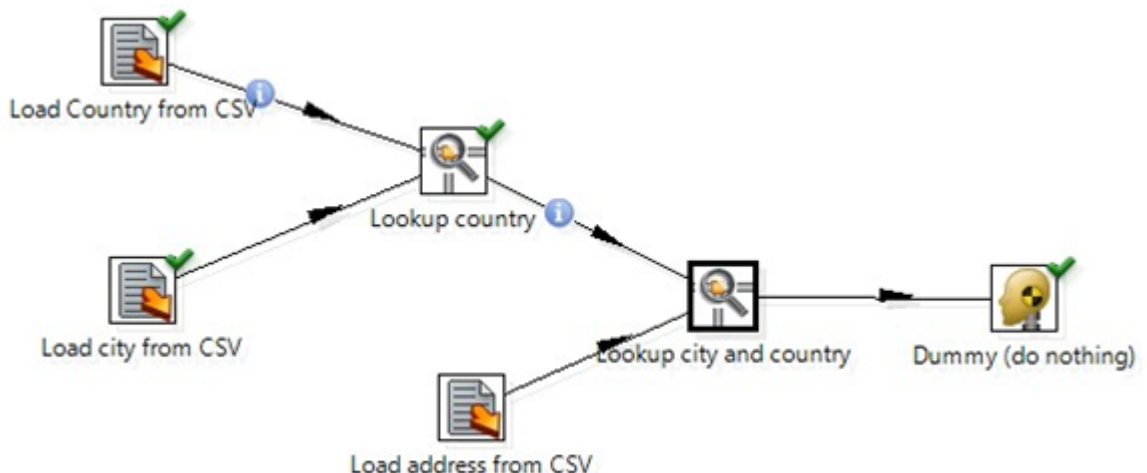
Preserve memory (costs CPU)

Key and value are exactly one integer

Use sorted list (i.s.o. hashtable)

Buttons: Help, OK, Cancel, Get Fields, Get lookup fields

Connect a hop from this new Stream lookup to the Dummy step. Our transformation should look something like this:





Save and launch the transformation and preview the results.

#	address_id	address	address2	district	city_id	postal_code	phone	city	country
1	1	47 MySakila Drive	<null>	Alberta	300	<null>	<null>	Lethbridge	Canada
2	2	28 MySQL Boulevard	<null>	QLD	576	<null>	<null>	Woodbridge	Australia
3	3	23 Workhaven Lane	<null>	Alberta	300	<null>	14033335568	Lethbridge	Canada
4	4	1411 Lillydale Drive	<null>	QLD	576	<null>	6172235589	Woodridge	Australia
5	5	1913 Hanoi Way	<null>	Nagasaki	463	35200	28303384290	Sasebo	Japan
6	6	1121 Loja Avenue	<null>	California	449	17886	838635286649	San Bernardino	United States
7	7	692 Joliet Street	<null>	Attika	38	83579	448477190408	Athenai	Greece
8	8	1566 Ineagl Manor	<null>	Mandalay	349	53561	705814003527	Myingyan	Myanmar
9	9	53 Idfu Parkway	<null>	Nantou	361	42399	10655648674	Nantou	Taiwan
10	10	1795 Santiago de Com...	<null>	Texas	295	18743	860452626434	Laredo	United States

Adding the address to the Customer table

Right-click on the hop from the lookup to the Dummy and select **Delete hop** from the menu.

Drag on a new **CSV file input** onto the transformation and configure it to load the [customer.csv](#) file and to load all of the fields from that file.

Drag on a new Stream lookup and connect hops from the previous lookup and from the customer CSV file input to this new lookup.

Edit the properties of this new lookup. Set the **Lookup step** to be the previous city and country lookup. The keys for this lookup will both be **address_id**. Click the **Get lookup fields** button to retrieve all nine of the fields from the address.

Connect this lookup step to the Dummy. Save and launch to test.



Combining transformations

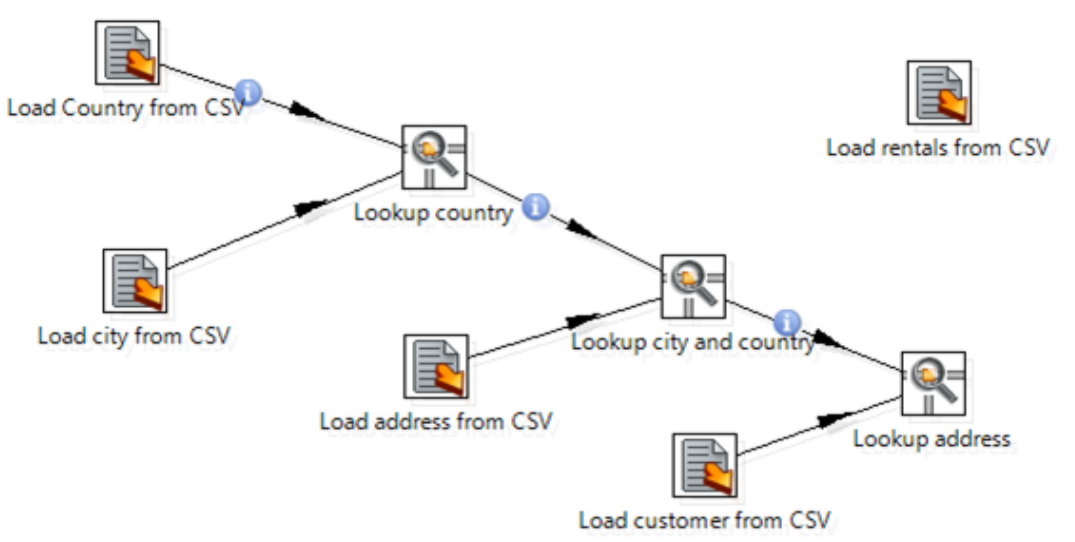
Quite often, we will want to bring together transformation steps from different transformations to create a larger transformation. An example of this is where we create a small set of transformation steps to test that a particular set of rules will work the way we expected; once we are happy, we can copy those steps into a larger transformation.

Gathering steps

Create a new transformation and save it as `Customer Rentals`.

Open (or re-open) the `My First Transformation` file. Click on just the `Load rentals from CSV` step and press `Ctrl-C` to copy it to the clipboard. Go to the `Customer Rentals` transformation and press `Ctrl-V` to paste the single step. Now, open (or re-open) the `Customer Transformation` file. Click on the transformation and drag across all of the steps, except for the `Dummy` step, to select them. Press `Ctrl+C` to copy the steps. Go to the `Customer Rentals` transformation, right-click on it and select **Paste from clipboard**.

It should now look like this screenshot:



Save `Customer Rentals`.



Save Customer Rentals.

Using Joins

We have already seen how to use the Stream lookup to combine data from different sources. That method can be used in many circumstances and is recommended if we are not sure what one to use. There are other methods of combining data and one of the main ones is to use joins – mostly the **Merge Join**.

Introducing joins

We can join two lots of data if they have one or more common fields (even if those fields do not have the same name). For example, in our case, we are going to join together the customer and rental tables based on them both having a `customer_id` fields.

There are several types of join. To explain them, we will use two simple tables of data:

Table1	
Key	Field1
1	A
2	B
3	C

Table2	
Key	Field2
2	X
3	Y
4	Z



Join type

Join description

INNER

The two tables of data are joined but only rows that have keys that exactly match from both sides are retained.

INNER JOIN		
Key	Field1	Field2
2	B	X
3	C	Y

LEFT OUTER

The two tables of data are joined. All rows from the LEFT table – the table that is listed first – are retained along with data that matches from the RIGHT table. If there is no key match with the right table then the field (in this case, Field2 for Key 1) will be Null, having no value.

LEFT OUTER JOIN		
Key	Field1	Field2
1	A	
2	B	X
3	C	Y

RIGHT OUTER

This is opposite to the LEFT. The rows from the RIGHT table are retained. Values for rows that don't match in the LEFT table will be Null.

RIGHT OUTER JOIN		
Key	Field1	Field2
2	B	X
3	C	Y
4		Z

FULL OUTER

In the FULL OUTER, all rows are retained. Fields will have Null where there are no key matches.

FULL OUTER JOIN		
Key	Field1	Field2
1	A	
2	B	X
3	C	Y
4		Z



In the majority of situations, either a LEFT JOIN or INNER JOIN are used.

Sorting the data

Before we can perform a Merge Join operation, we need to explicitly sort the input rows on the key fields that are going to be used in the join. In our case, both outputs will be sorted on the `customer_id` field before inputting to the Merge Join.

To sort the data, we need to add two **Sort rows** tools, from the **Transformation** folder in the **Design** tab, onto the transformation. Add a hop from the **Load Rentals from CSV** step to one of the Sort rows steps (**Main output**). Add a hop from the **Lookup address** step to the other Sort rows step.

Edit the first Sort rows step. We need to specify the **Fieldname** to sort by to be `customer_id`. We should also specify **Ascending** (the order of the sort – ascending means A to Z and 0 to 9) to be **Y**. For the purpose of sorting for a Merge Join, the sort must always be ascending.

Sort rows

Step name: Sort rentals by customer_id

Sort directory: %%java.io.tmpdir%% [Browse...](#)

TMP-file prefix: out

Sort size (rows in memory): 1000000

Free memory threshold (in %):

Compress TMP Files?

Only pass unique rows? (verifies keys)

Fields:

#	Fieldname	Ascending	Case sensitive compare?	Presorted?
1	customer_id	Y		

Help OK Cancel Get Fields



Edit the properties for the second Sort rows step and set the same **Fieldname** and **Ascending** – `customer_id` and **Y**.

Creating the join

Add a **Merge Join** tools, from the **Joins** folder in the **Design** tab, onto the transformation. Connect a hop from both of the sort steps to the Merge Join. Edit the properties:

#	Key field	
1	customer_id	

#	Key field	
1	customer_id	

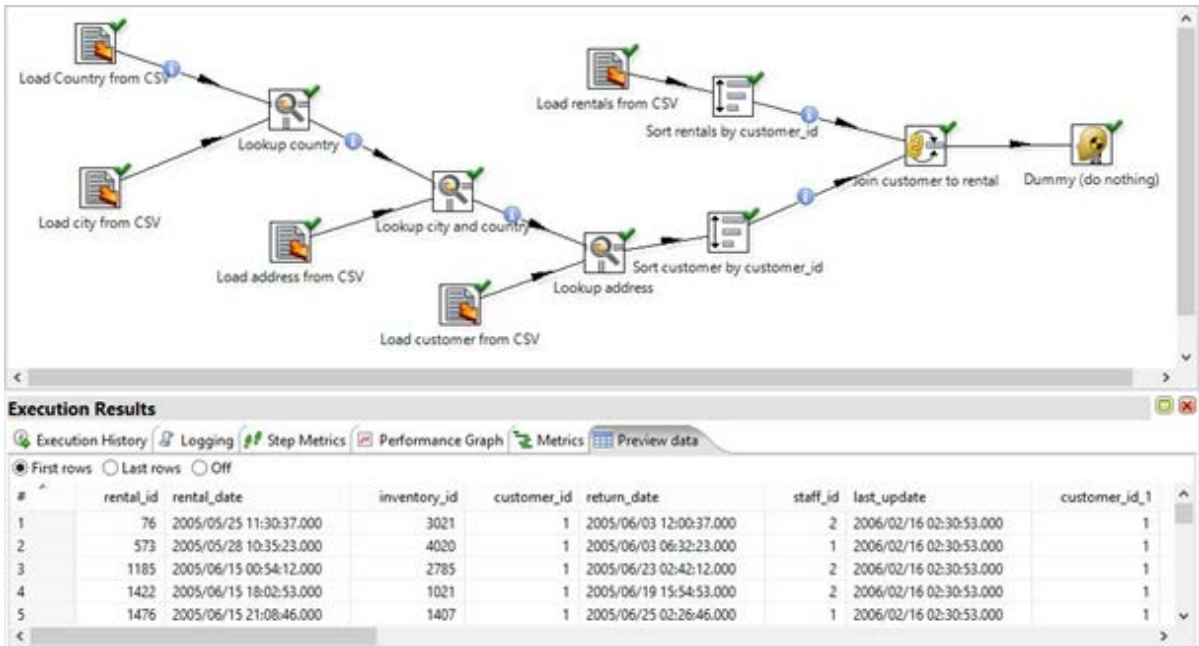
We are going to start with the rentals step as the **LEFT** side of this join and the customer step will be the right. Rentals will be retained whether there is a match or not. Customers who have never rented will be lost.

Set the **First Step** to be the rentals sort. Set the **Second Step** to be the customers sort. Type `customer_id` in for both **Key field** entries.

*You can use the **Get key fields** button, but it does return all fields whether they are keys or not – it is often quicker just to type the fields in here.*

Add a Dummy step to receive the output of the join.

Save the transformation and test it. Review the Preview data.



The screenshot displays a data integration workflow in a tool like Talend. The workflow consists of several steps: 'Load Country from CSV', 'Load city from CSV', 'Load address from CSV', 'Load customer from CSV', 'Load rentals from CSV', 'Lookup country', 'Lookup city and country', 'Lookup address', 'Sort customer by customer_id', 'Sort rentals by customer_id', 'Join customer to rental', and 'Dummy (do nothing)'. Below the workflow, the 'Execution Results' section is visible, showing a table of data with columns: #, rental_id, rental_date, inventory_id, customer_id, return_date, staff_id, last_update, and customer_id_1. The table contains 5 rows of data.

#	rental_id	rental_date	inventory_id	customer_id	return_date	staff_id	last_update	customer_id_1
1	76	2005/05/25 11:30:37.000	3021	1	2005/06/03 12:00:37.000	2	2006/02/16 02:30:53.000	1
2	573	2005/05/28 10:35:23.000	4020	1	2005/06/03 06:32:23.000	1	2006/02/16 02:30:53.000	1
3	1185	2005/06/15 00:54:12.000	2785	1	2005/06/23 02:42:12.000	2	2006/02/16 02:30:53.000	1
4	1422	2005/06/15 18:02:53.000	1021	1	2005/06/19 15:54:53.000	2	2006/02/16 02:30:53.000	1
5	1476	2005/06/15 21:08:46.000	1407	1	2005/06/25 02:26:46.000	1	2006/02/16 02:30:53.000	1

Summary

In this chapter we have met the very useful Stream lookup tool – which we will use again and again for straightforward blending of data. We have seen how we can combine several steps to blend data together from multiple sources.

We introduced the concept of joining data and explained the different types of joins. We showed how to sort data in preparation for a Merge Join operation and learned that it is a mandatory step.

In the next chapter we will learn how to create calculated fields using our data so that we can further extend the model.



CHAPTER FOUR

MODIFYING DATA



Modifying Data

The power of a tool like Kettle is not just confined to loading data from multiple sources, blending it, then writing out into a format that we can consume. There is also a whole suite of tools available to manipulate that data.

For example, we can take an existing field, for example a date, and split it out into fields that will be useful for analysis, like year, month and day. We can derive new fields based on the condition of another field or by combining multiple other fields.

In this chapter we will introduce the Calculator tool – an easy to use tool to perform calculations on existing fields. We will also have look at the Formula tool which allows us to create advanced expressions. Finally, we will consider the possibilities of creating new data, for example for a lookup table, whenever we might need it.

These are the topics we'll be covering in this chapter:

- Using Calculator to derive calendar fields
- Examining the Formula tool
- Creating new data



Using Calculator to derive calendar fields

In almost every analysis that we perform, there will be the requirement to look at the data by date, not just by individual days, but grouping into date segments like Year and Month. In this section, we will use the Calculator tool to derive those fields from the rental date.

Introducing the Calculator tool

The Calculator tool is used to create new fields by running simple calculations. We can pass up to three fields as parameters to the various available functions. However, we do not have fully flexible control on each calculation – we need to accept the functions that are available. Often that is good enough for most purposes.

Some of the types of calculations that we can perform are:

- Create a constant field. For example, create a field to hold the number 1 so that we can sum it.
- Simple calculations – $A+B$, $A*B$, etc.
- Calendar calculations – extract Year, Month and Day from a data.
- String calculations – parse and format text.

Performing the calendar calculation

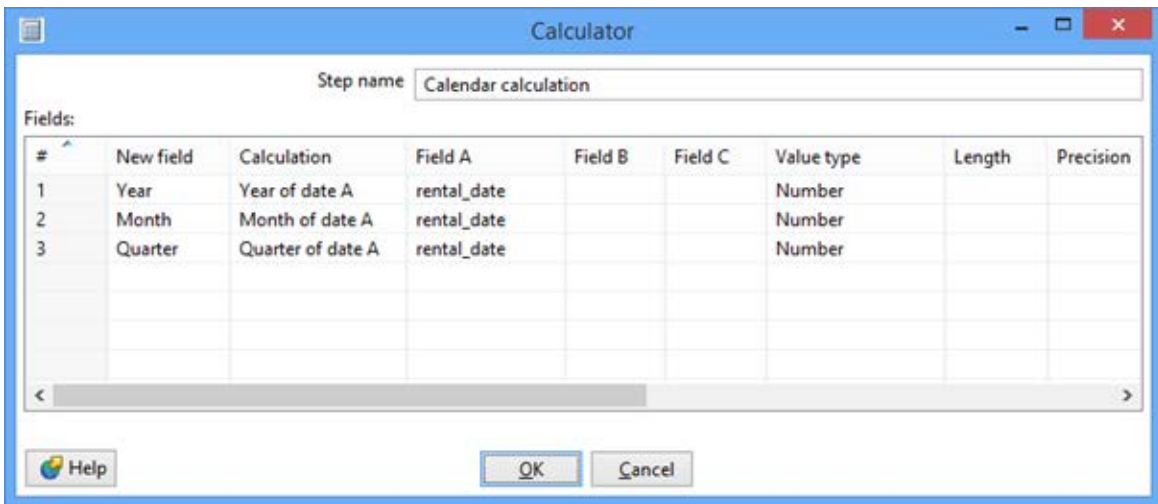
First, we should save our existing Customer Rentals transformation as Customer Rentals with Calculations. Delete the hop between the Merge Join and the Dummy.

From the Transform folder in the Design tab, drag a Calculator tool onto the transformation. Edit the properties. In the first row, enter **Year** as the **New field** name. Click on the **Calculation** field and the **Select the calculation type** dialog box appears.



Click into line 2 and enter **Month** for the **New field**. Click into the **Calculation** and select **Month of date A** for the calculation type. Again, select rental_date for Field A and set the **Value type** of **Number**.

In line 3, we will create a **Quarter** field and use the **Quarter of date A** function. When complete, the Calculator dialog should look like this:



Click **OK** to save the settings. Connect a hop from the calculator step to the Dummy step. Save the transformation and test it. Preview the data to see the newly created fields. Note that each new value has one decimal place displayed, even though they are integers.

Edit the calculator step again. In the **Conversion mask** field for each field, select **#** from the dropdown. Save and retest the transformation. Notice that the decimal place has disappeared.



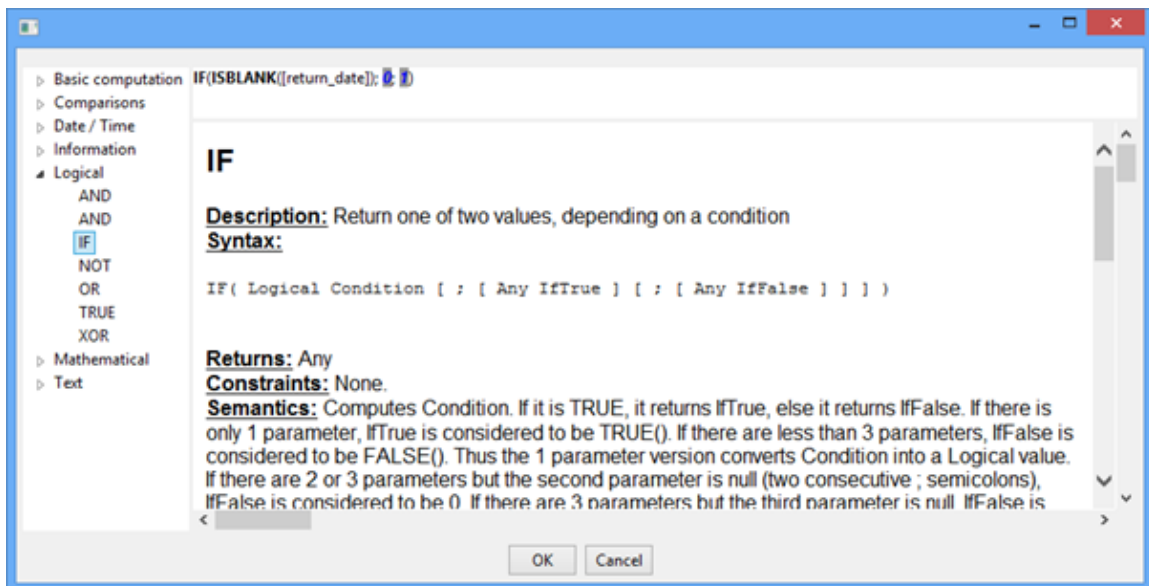
Examining the formula tool

We sometimes have to go a little beyond the calculations that are contained in the Calculator tool. That is when we turn to the formula tool to give us full power.

Introducing the formula tool

The formula tool allows us to enter a complex calculation – although the syntax of many of the functions are very similar to those found in Excel, so there may be some familiarity already.

The interface is quite simple. On the left side there is a list of the most common functions that we will use most frequently. When we click on one of the functions, the help for that function appears in the large panel on the right. The smaller top panel on the right is where we enter our function.





Performing a calculation with Formula

We want to create a flag field to indicate if a rental has been returned or not. We will judge this by looking at the `return_date` field and seeing if it is blank or not. If it is populated then the rental has been returned. If it is blank, then the rental is still outstanding.

Again, we will break the hop leading to the Dummy step. From the **Scripting** folder in the **Design** tab, drag a **Formula** tool onto the transformation. Create a hop from the Calculator step to the Formula step. Edit the Formula step.

In the first line, click into **New field** and enter **flag_is returned**. Click into the calculation and the formula dialog appears. Enter the formula:

```
IF(ISBLANK([return_date]); 0; 1)
```

Click **OK**. Click **OK** on the **Formula** dialog. Connect a hop from the Formula step to the Dummy step. Save the transformation and launch to test.

Note that the IF function is very like the Excel equivalent. The notable difference is that semi-colons are used instead of commas. Also note that field names are identified by enclosing them in square braces.

Creating new data

It is quite often necessary for us to add data to a transformation that does not exist anywhere else. For example, we may need to create a lookup table that exists in an application but not in a database.

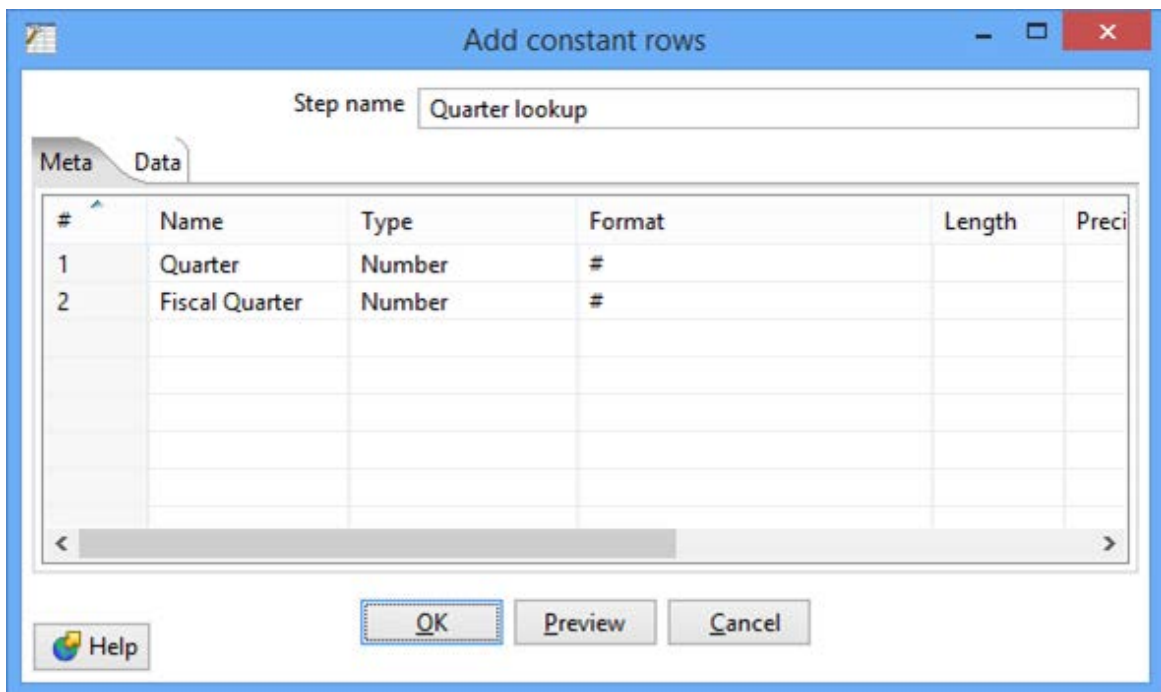
The tool that we use in Kettle for this task is the Data Grid.



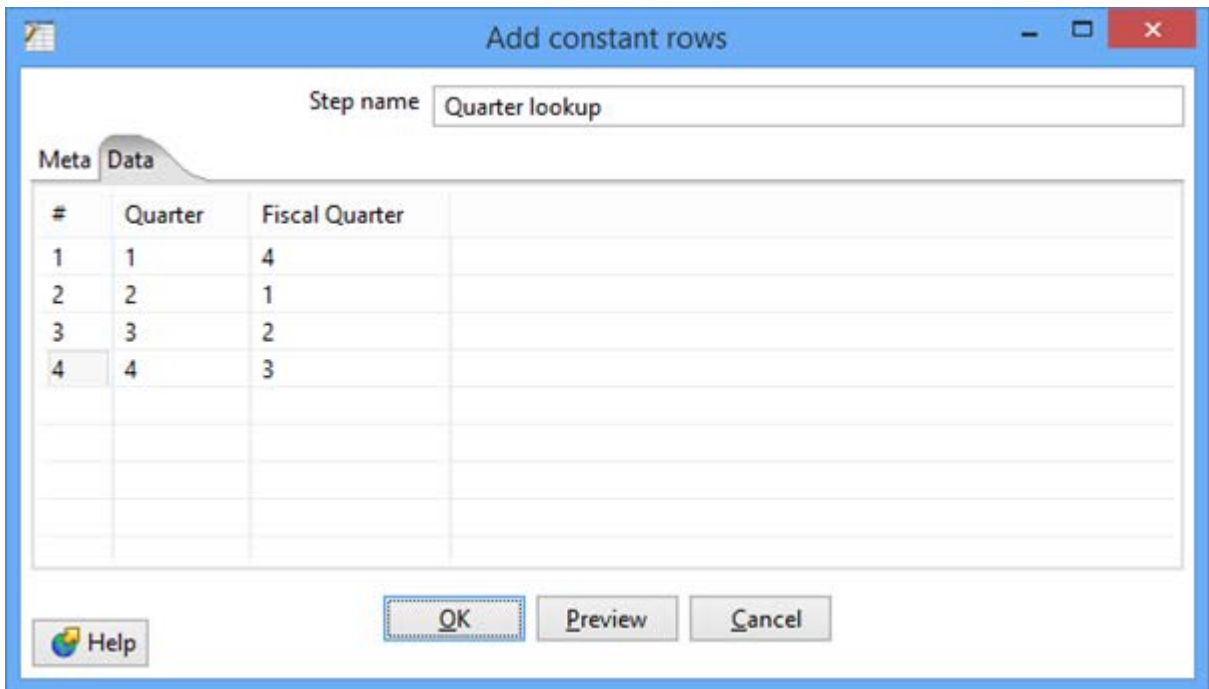
Using the Data Grid

We want to use the data grid to create a Fiscal Quarter field to reflect that the company starts its year in April instead of January. To do that, we are going to map from the calendar quarter field that we have already calculated to a table that we will enter in a Data Grid. We will then use a Stream lookup to add this field to our data.

First, we can now delete the Dummy step as we will not be using it again. In the **Design** tab, under the **Input** folder, drag a **Data Grid** tool onto the transformation. Edit the Data Grid properties.



Add two **Number** fields – **Quarter** and **Fiscal Quarter**. The **Format** should be #. Click onto the **Data** tab.



Enter the data as shown in the screenshot above. Quarter 1 maps to Fiscal Quarter 4, Quarter 2 to Fiscal Quarter 1, etc. Click **OK** when done.

Drag a Stream lookup onto the transformation and connect hops to it from both the Data Grid step and the Formula step. Edit the properties of the Stream lookup.

Set the **Lookup step** to the quarter table that we have just created. The **Field** and **LookupField** are both **Quarter**. We only want one **Field** retrieved – **Fiscal Quarter**. The settings should look similar to this:



Stream Value Lookup

Step name: Lookup fiscal quarter

Lookup step: Quarter lookup

The key(s) to look up the value(s):

#	Field	LookupField
1	Quarter	Quarter

Specify the fields to retrieve:

#	Field	New name	Default	Type
1	Fiscal Quarter			Integer

Preserve memory (costs CPU)

Key and value are exactly one integer field

Use sorted list (i.s.o. hashtable)

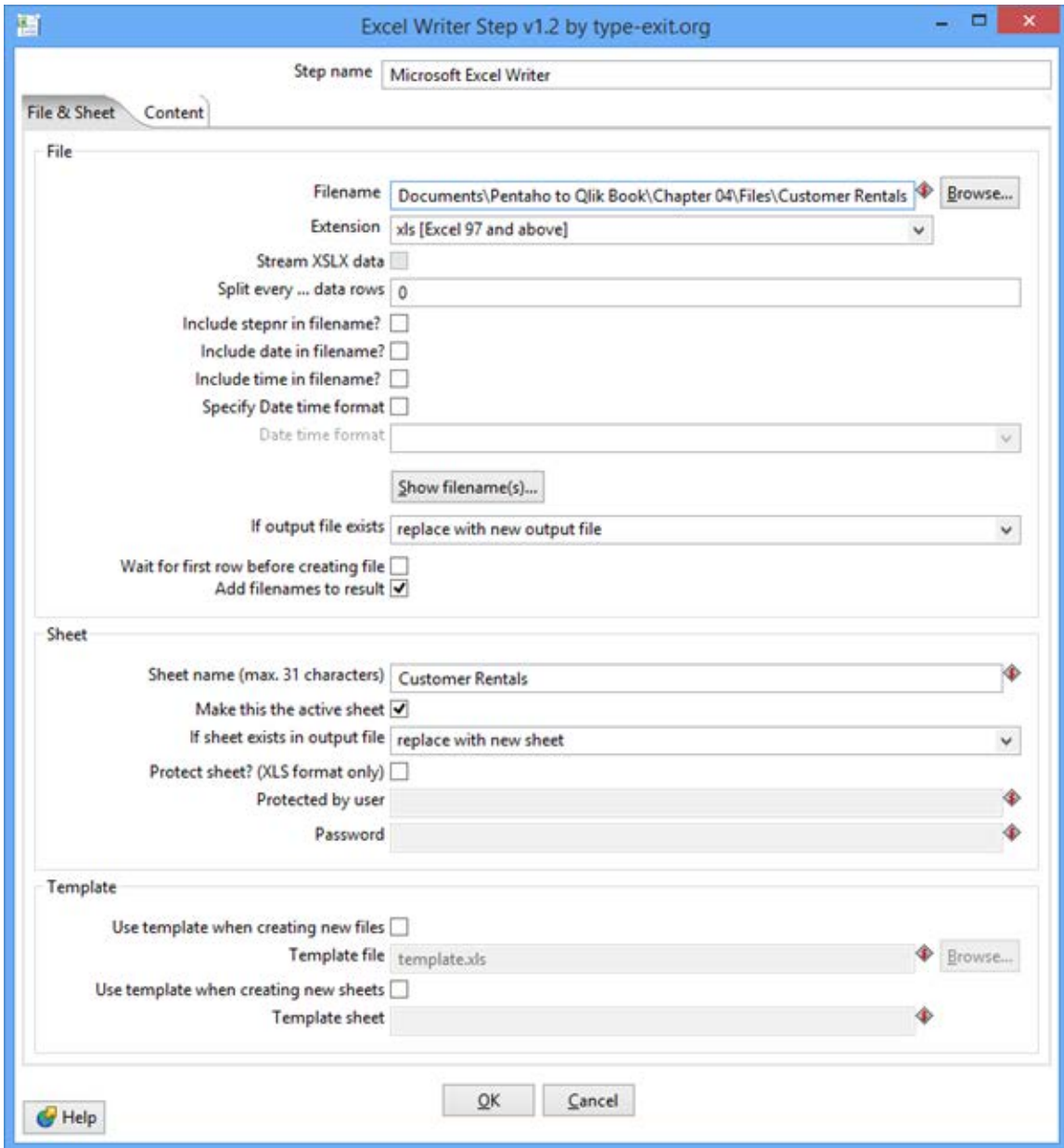
Buttons: Help, OK, Cancel, Get Fields, Get lookup fields

We are almost done creating our data!

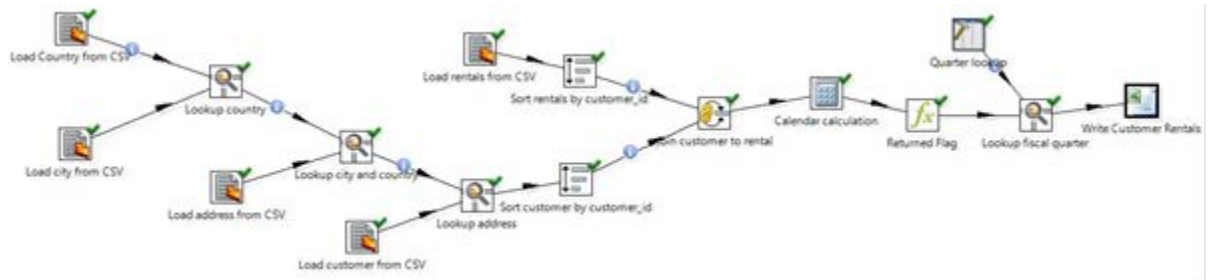
Writing our data to Excel

We now need to take all the data that we have transformed and write it to Excel.

As we did before, from the **Output** folder of the **Design** tab, drag a **Microsoft Excel Writer** tool onto the transformation. Edit the properties.



Configure the **Filename** to write the Customer Rentals xls file to an appropriate folder. Click on the **Content** tab and click on **Get Fields** to populate the fields to export. Click OK. The transformation should look something like this:



Save the file and launch the transformation to confirm that the file is written correctly.

Summary

In this chapter we discovered how we can calculate new fields using the simple Calculator tool and the more advanced Formula tool. We then saw how we could supplement the data from external sources with a static table that we enter in the transformation.

In the next chapter we will take the file that we have exported to an Excel format and load it into Qlik Sense and start analysing.



CHAPTER FIVE

DISCOVERING WITH QLIK SENSE DESKTOP



Discovering with Qlik Sense Desktop

We have spent the last few chapters spending time in manipulating the data with Pentaho Kettle. Now the fun begins and we can load it into Qlik Sense to start analyzing and discovering.

One thing that we will discover is just how easy it is to get started with creating applications in Qlik Sense Desktop – in fact, it is as easy as just drag and drop! These are the topics we'll be covering in this chapter:

- Creating a new app in Qlik Sense Desktop
- Loading data from files
- Creating our first analysis
- Using images and text
- Telling stories
- Sharing apps with other users



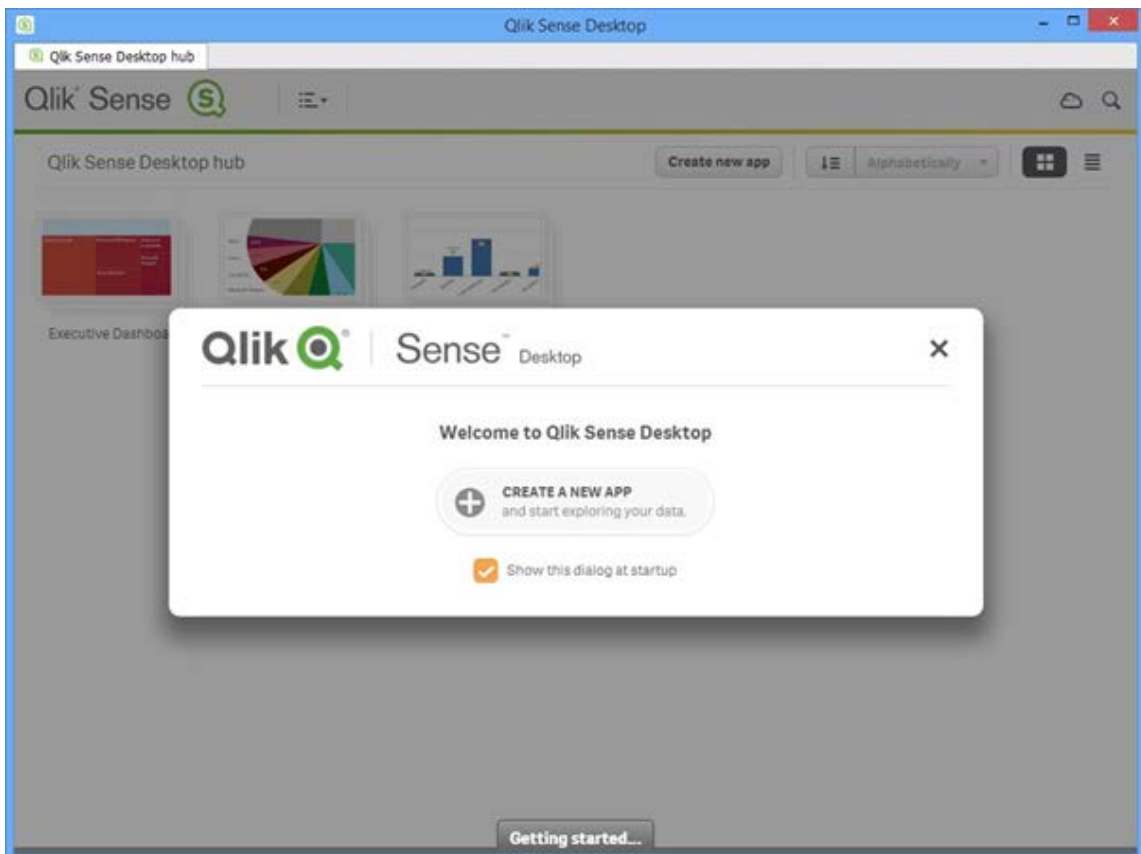
Creating a new app in Qlik Sense Desktop

When creating their new product, Qlik spent a lot of time on thinking about how users might use Qlik

Sense to create applications and they have made it really very simple.

Starting Qlik Sense Desktop

The first step is to actually launch Qlik Sense Desktop. We do this by double-clicking the desktop icon. We are greeted with the **Welcome to Qlik Sense Desktop** dialog:



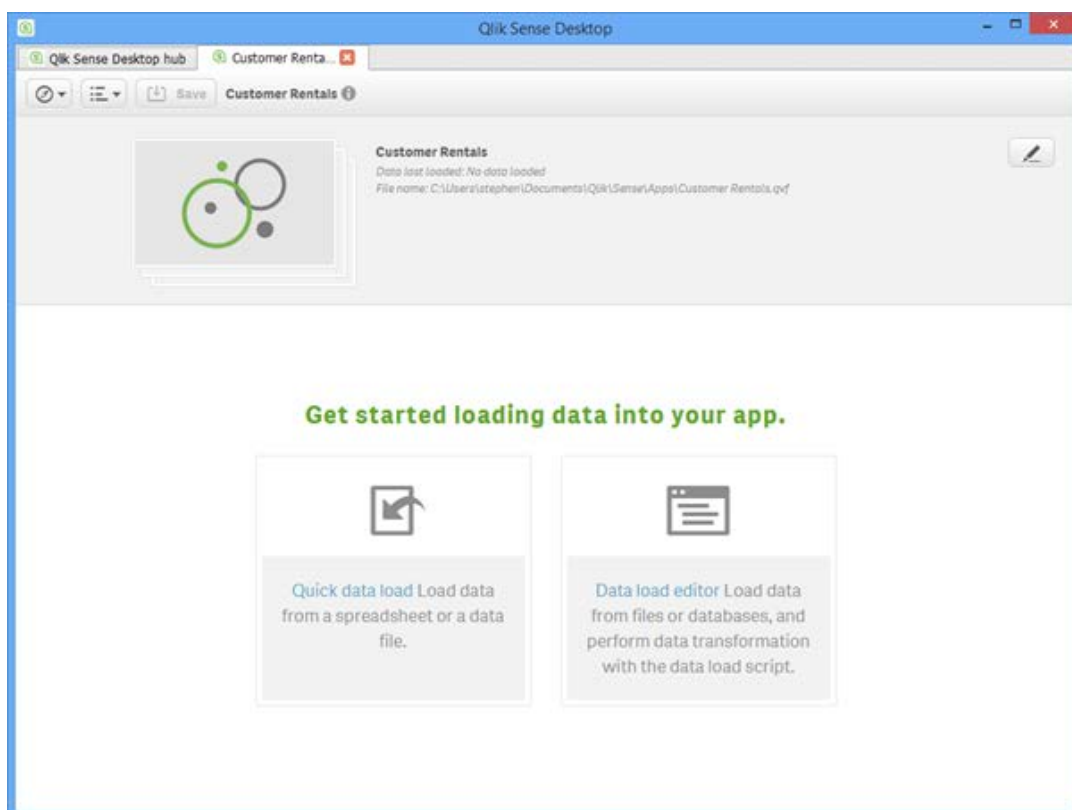


Creating a new app

Rather invitingly, this dialog has a very large button that invites us to **CREATE A NEW APP** and **start exploring your data**. Let's click this button.

*If the **Welcome** dialog does not appear, perhaps because you have previously deselected the **Show this dialog at startup** checkbox, you can simply click the **Create new app** button in the upper right of the **Qlik Sense Desktop hub**.*

We should give the application an appropriate name. For example, Customer Rentals would reflect the data that we are going to put in it. Enter the name and click on the **Create** button. When the **New app created** dialog appears, click on **Open app**. A new tab is opened in Qlik Sense Desktop to display the app. Note that the hub is still available on its own tab.



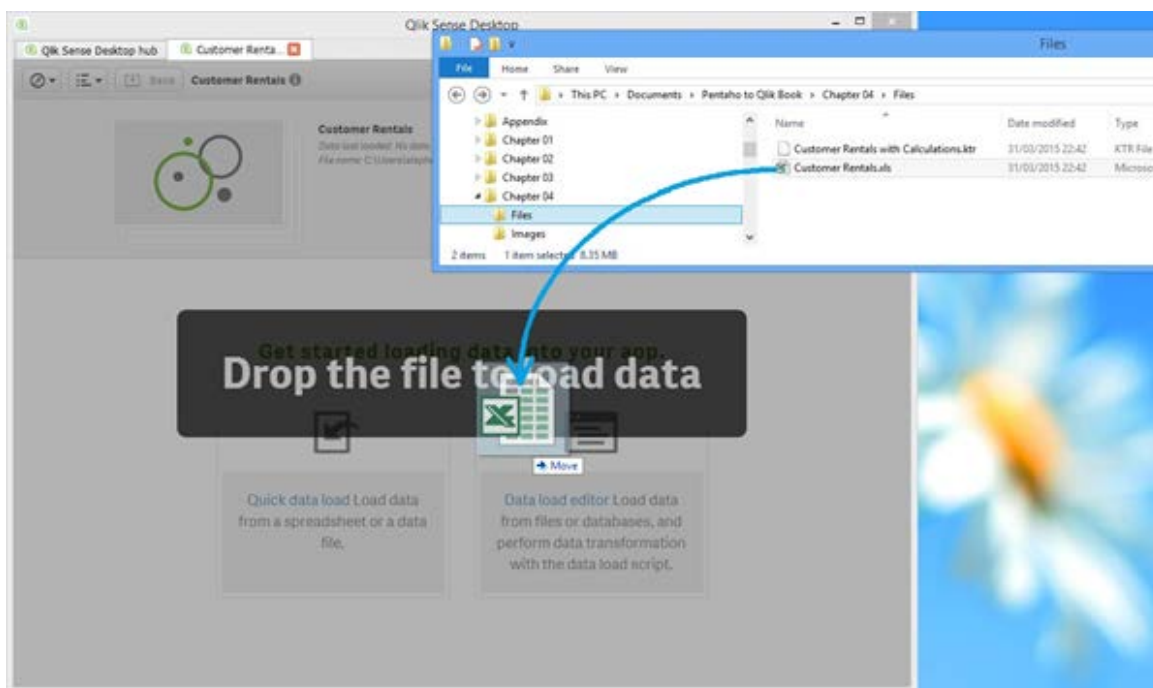


Loading data from files

There are two large buttons on the **Getting started** screen that allow us to load data. Probably, or until we have more experience with the product, we won't need to use the **Data load editor** at all. This gives access to the Qlik Sense data scripting language – it is extremely powerful, but not for the inexperienced. And we don't have to use it as we have already done all the hard work in Kettle!

The Quick data load option comes in two flavours. We can either click on this button and then browse to a file. Alternatively (and this is my favourite!), we can just drag a file onto the Qlik Sense Desktop!

Let's try that – locate the file that we created in the last chapter – [Customer Rentals.xls](#). Then simply click on it and drag it onto the Qlik Sense Desktop and release:



The **Select Data** wizard will appear:



Select data from Customer Rentals.xls

Tables:

File format: Excel (XLS)

Field names: Embedded field names

Header size: - 0 +

Fields:

rental...	rental_date	inventory...	customer...	return_c
76	38497.479594907	3021	1	38506.500428
573	38500.441238426	4020	1	38506.272488
1185	38518.037638889	2785	1	38526.112638
1422	38518.752002315	1021	1	38522.663113
1476	38518.881087963	1407	1	38528.101921
1725	38519.638159722	726	1	38520.879131
2308	38521.362361111	197	1	38525.150558
2363	38521.565266204	3497	1	38522.736793
3284	38524.2671875	4566	1	38531.144968
4526	38541.136863426	1443	1	38547.054918
4611	38541.315231481	3486	1	38545.559675
5244	38547.880418957	3778	1	38547.804108

Cancel Load data

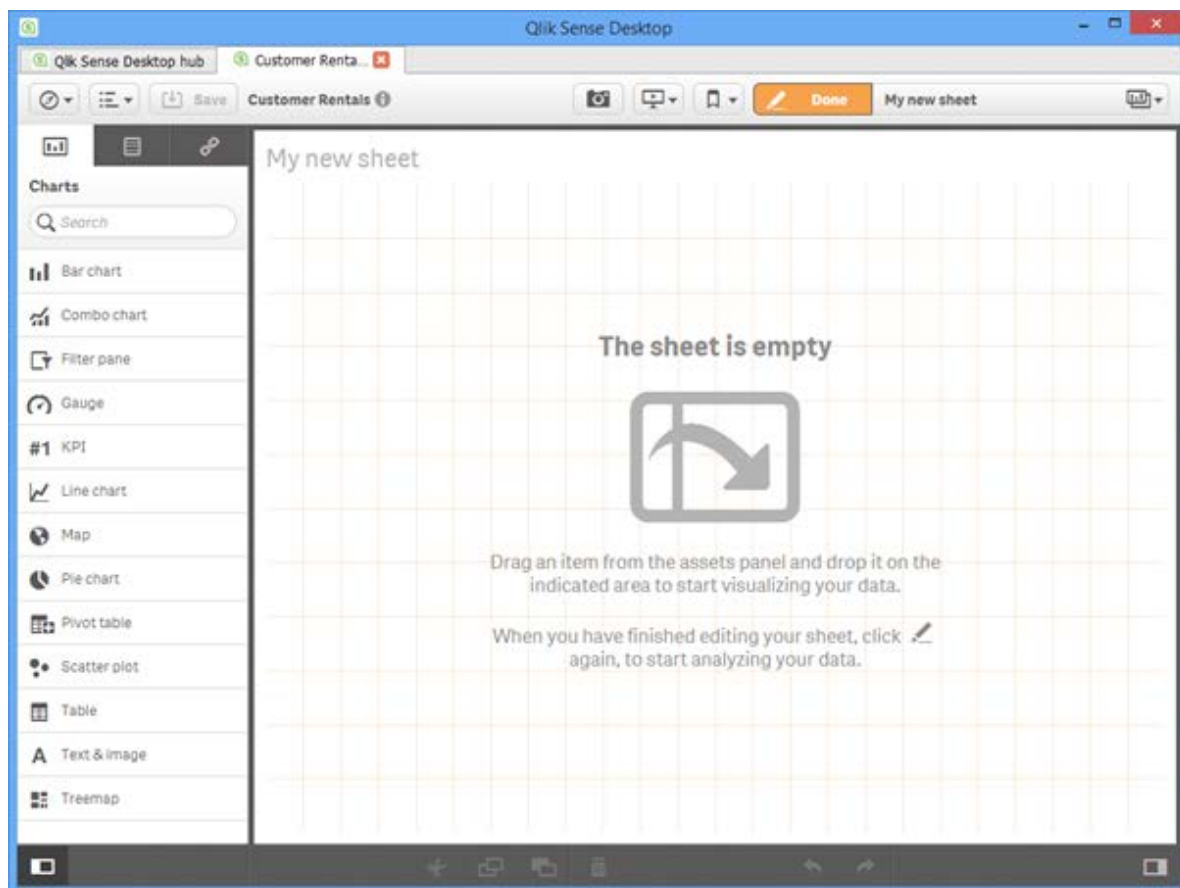
In this wizard, we can select multiple tables of data in the file to load at the same time, we can rename fields, and we can deselect fields that we don't want to load.

At this stage, we should click the **Load data** button. Qlik Sense Desktop will load the data and create a new, blank, sheet for us to start working with.

Creating our first analysis

Once the data has been loaded successfully, we can now edit the sheet that has been created for us. Click the **Edit the sheet** button or, if that dialog has been closed, just click on the sheet in the **App Overview** and click the **Edit** button.

*Now would be a good time to hit the **Save** button. Always a good idea to save as you go along.*



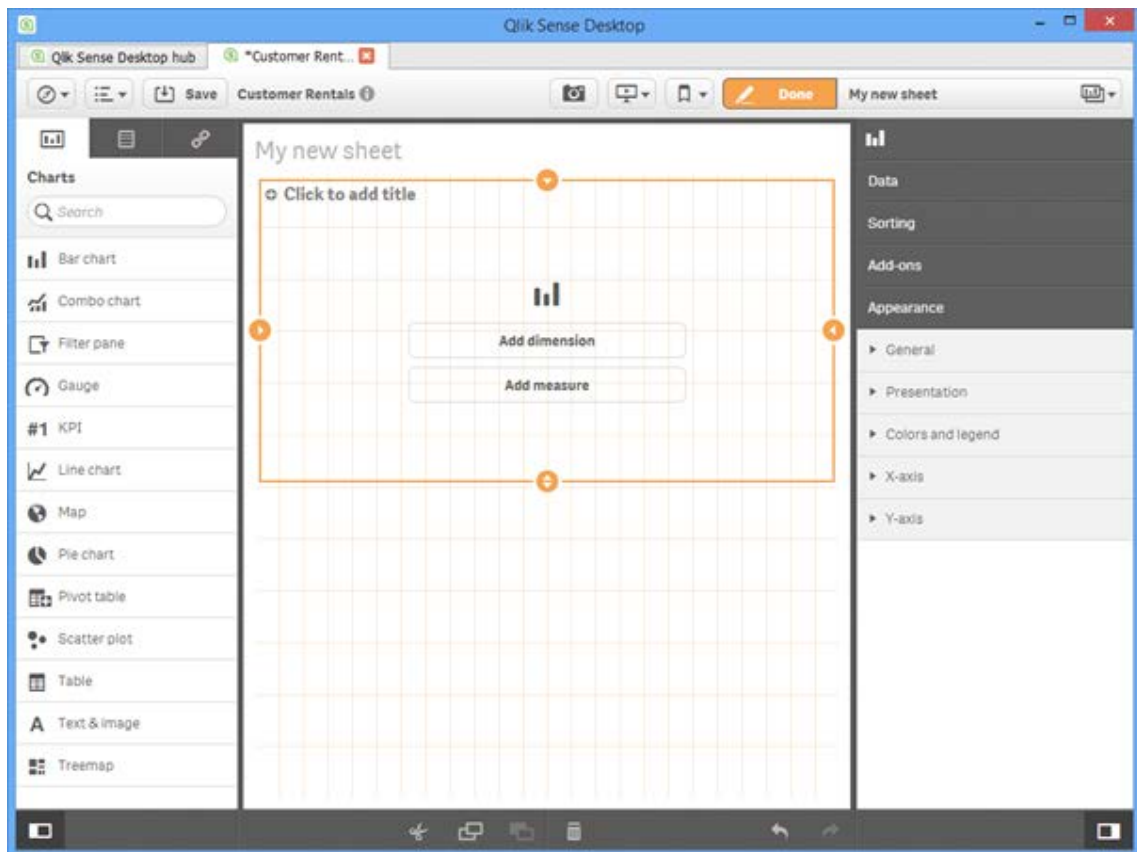
Adding charts and tables

The Assets panel on the left hand side has three tabs:

Tab	Description
Charts	These are all the different visualizations that you can add to the sheet.
Fields	This tab lists all the fields that have been loaded from the data source.
Master Items	If we want, we can identify particular fields as Dimensions and we can create particular expressions as Measures. These appear as Master Items. We can also save our favourite charts as Master Items.



Let's start by dragging a **Bar chart** from the charts panel onto the sheet. Before you drop the chart onto the sheet, notice the shading that appears when you move the mouse to different regions. This shading indicates where the object will appear once dropped.



Notice that each side of the object has drag-handles that allow us to resize. Notice also that a new panel has appeared – **Properties**, where we can tweak some settings if we need to. Note also the icons at the bottom to copy, paste, delete, and undo.

Let's click on the **Add dimension** button. All of our fields are listed. Select **country** from the list.

Now let's click on the Add measure button. Select rental_id from the list and, in the **Aggregation** dialog, select **Count(rental_id)**.



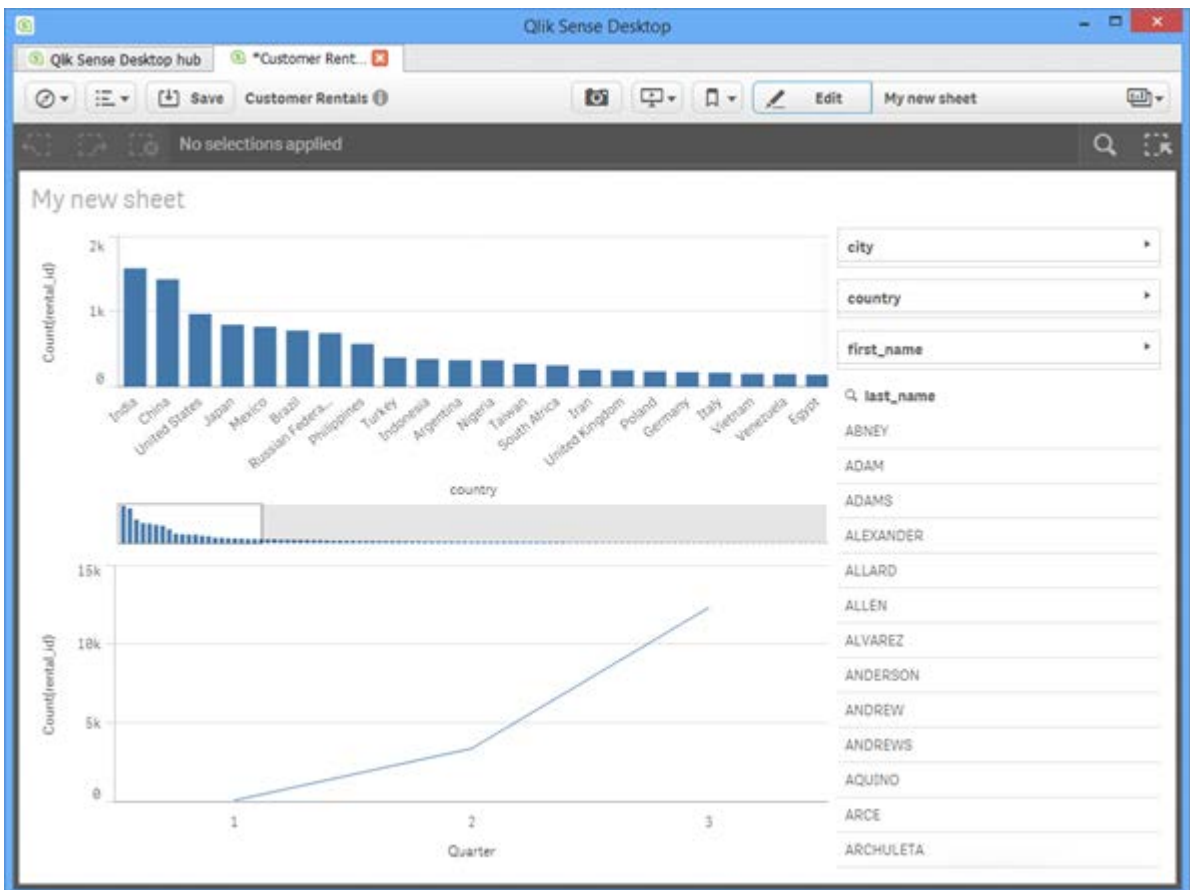
Now, drag a **Line chart** onto the sheet. Set the dimension to **Month** (try using the filter to search for it) and use the same measure as above - **Count(rental_id)**.

Go to the **Fields** tab in the **Assets** panel and drag the **Quarter** field onto the Line chart. Select **Replace "Month"** from the menu.

From the **Assets** panel, **Charts** tab, drag a **Filter pane** onto the sheet and position it on the right hand side. In the **Properties** panel under **Data**, click the **Add dimension** button. Add the fields, **city**, **country**, **first_name** and **last_name**.

Again, we could have done the same task by dragging fields from the **Fields** tab.

Click **Done** on the toolbar.





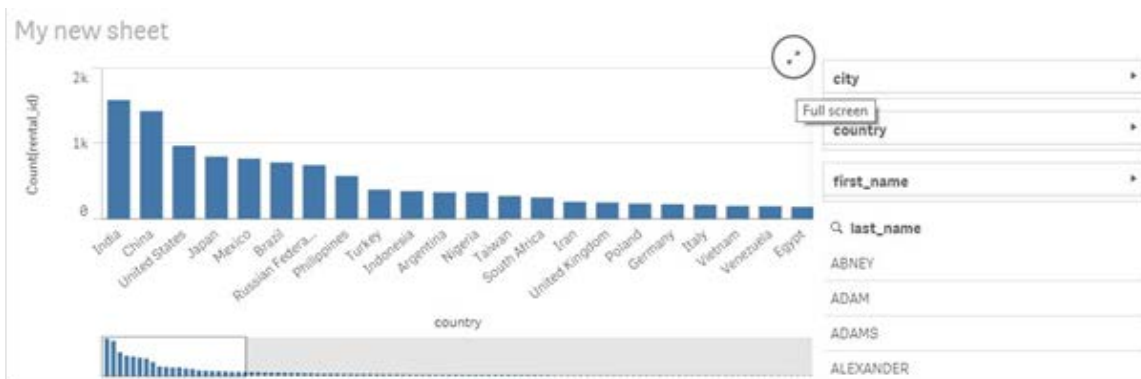
Exploring the app

We have created the app, but the charts that are on the screen are not just static displays of data. Everything in Qlik Sense Desktop is interactive and clickable (or touchable on a touch enabled screen).

Making things bigger

Depending on the screen size – and Qlik Sense will automatically resize objects based on the available size – some objects may be a little compressed and not showing as much detail as we would like to see. We can, however, quickly and easily expand any object to fill the screen.

When we hover over any object on the screen a **Full Screen** button appears in the top-right corner of the object:



If you click on the button, the object will expand to fill the whole screen. While in full screen mode, there is an **X** button in the top right corner that will restore the object to normal size.

Selecting data in filters

Using the filters that we added (city, country, first_name and last_name), we can click on items in the list and that will start to make selections:

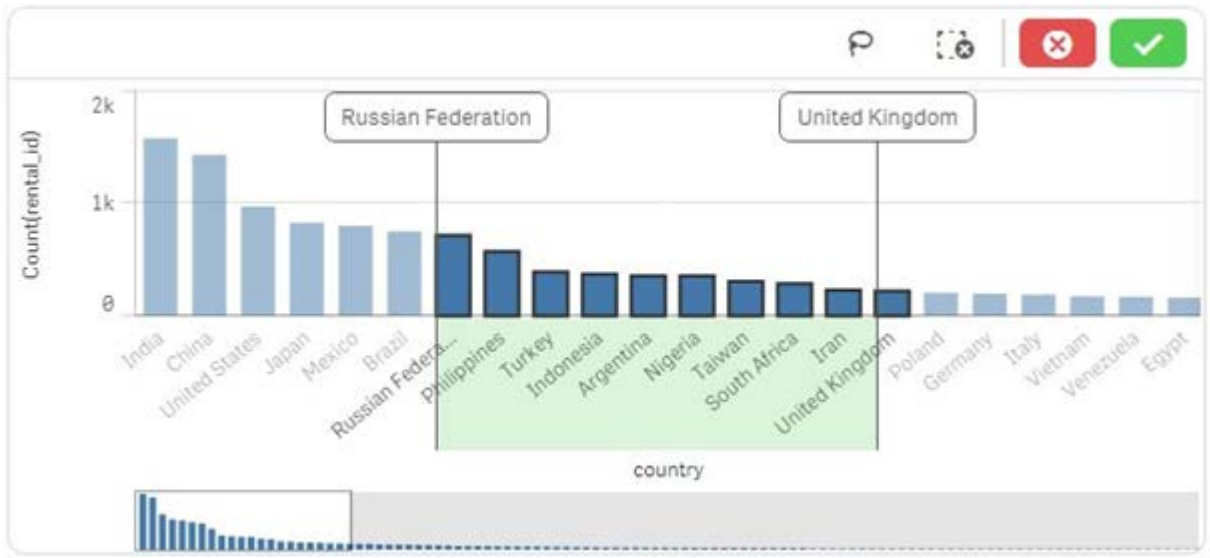


Notice that as we make selections in the filters, the other objects on the screen will automatically re- calculate to show you the results based on the filter values. Clicking the green **Confirm selection** button will confirm the selection. Clicking the **Cancel selection** button will cancel it.

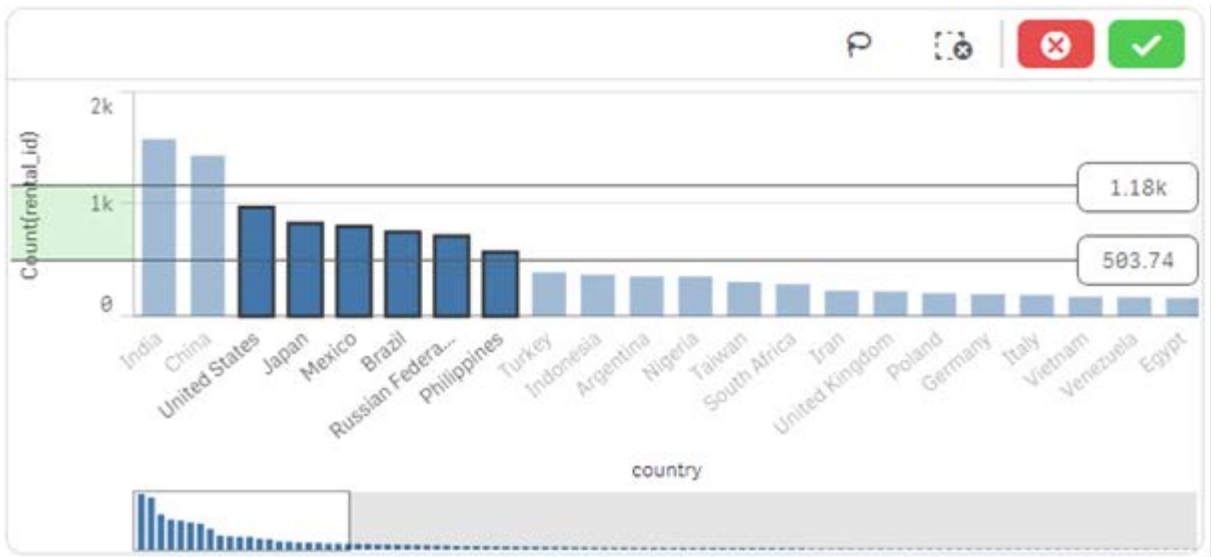
If the filter does not have its values displayed because of space, you can click on the filter and a popup will appear allowing you to make the selections. We can also expand the whole filter pane to fill the whole screen and see all the values.

Making selections in charts

All charts are selectable. For example, in a bar chart, we can click on individual bars to make selections. We can also click and drag along the dimension axis to select values:



We can also select on the measure axis to select dimensions based on the measure value:

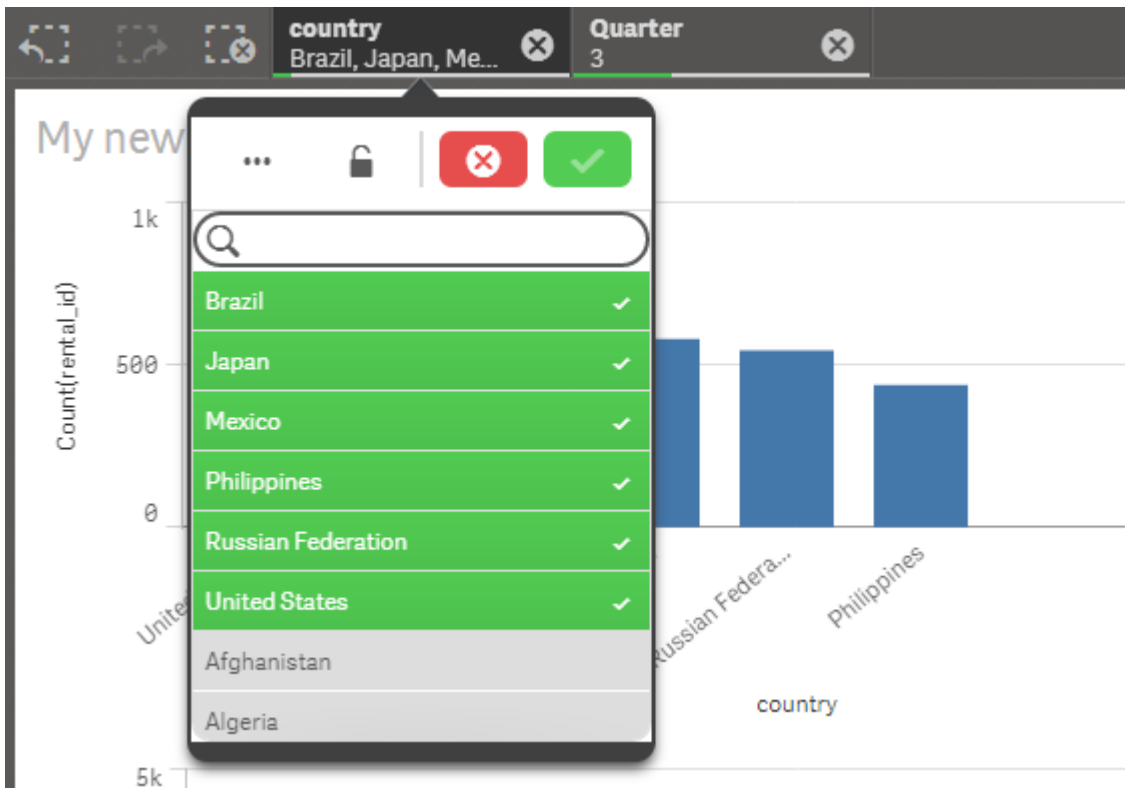


The selections are not confirmed until we click the **Confirm selections** button, but other objects will update to show the effect of the selections, even before the selection is confirmed.



Using the selections bar

The selections bar sits at the top of the sheet, showing us what selections have been made.



For each field that has a selection made, those selections are displayed under the field name. If there are more than a couple of selections, the list is truncated for space, but you can see them all by hovering over the fieldname. Clicking on the field will pop up a filter box to allow us to make different selections. We can also click the X button to the right of the field name to clear all the selections in that field.



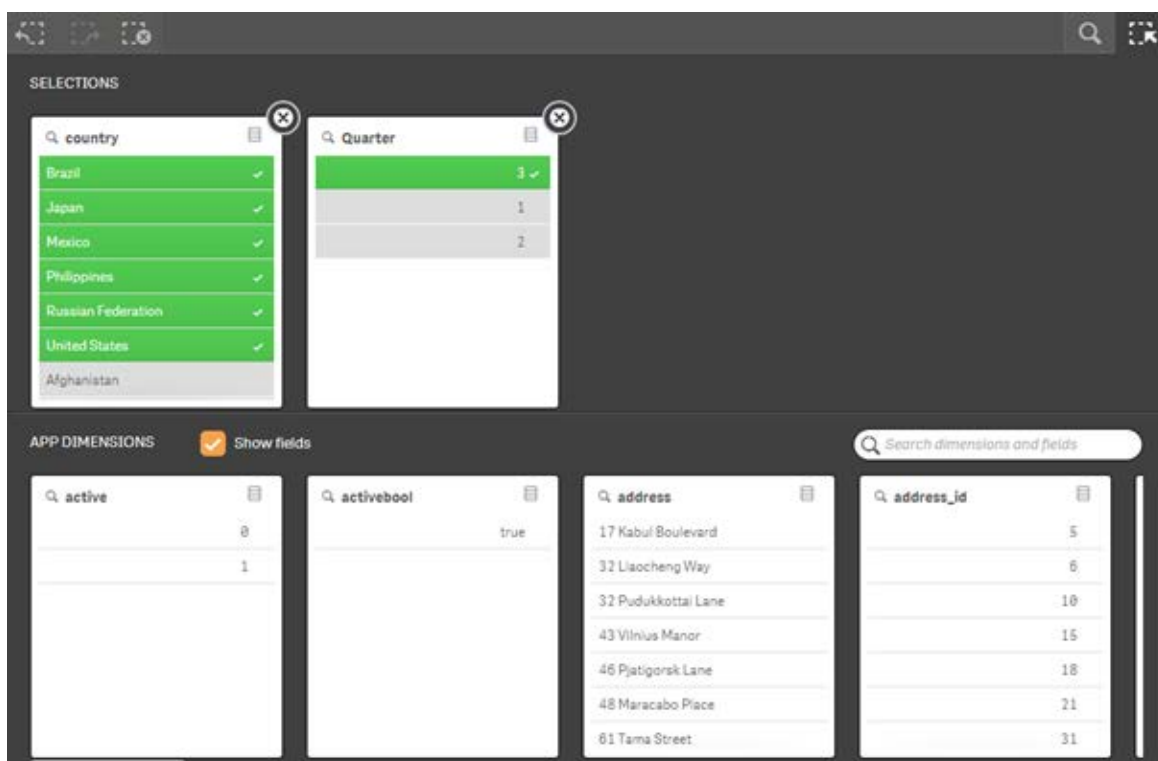
The first three buttons on the selections bar are very useful when navigating our data. They are:

- **Step back** – go backwards through recent selections.
- **Step forward** – if we have gone back, we can go forward again.
- **Clear all selections** – this clears selections in every field.

At the far right of the selection bar, there are two buttons that are very useful – **Selections tool** and **Search tool**.

Using the Selections tool

The **Selections tool** expands the selections bar to show us a filter panel for each field that has a selection on. Further, it shows us all the available fields in the app so that we can make new selections:

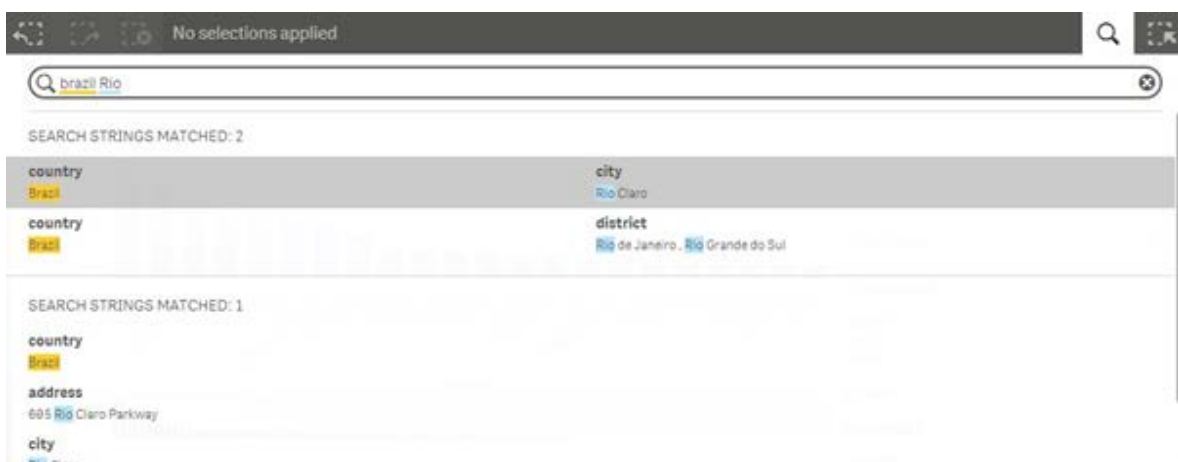




Searching associatively with the Search tool

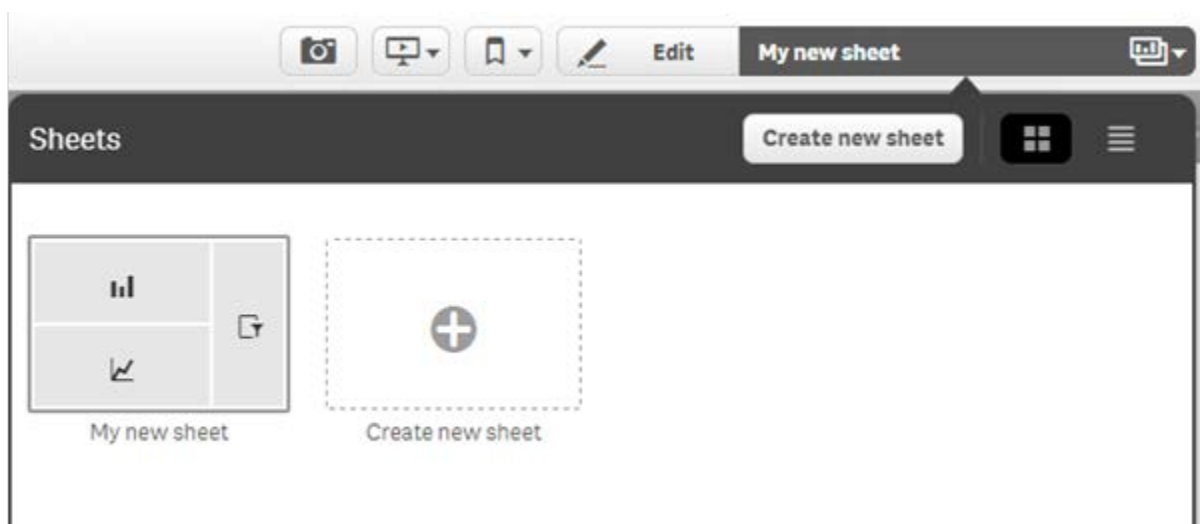
One of the most powerful features of Qlik Sense is the search function. The Search tool is a really innovative and brilliant tool to allow us to find information.

It allows us to enter multiple search terms and will look for records where it can find all the terms, but will also give us option where it has located some but not all:



Creating and navigating sheets

To the right of the Edit button, there is a button containing the name of the current sheet. Clicking on this button will bring up the **Sheets** dialog showing all the sheets in the app:





Creating a new sheet

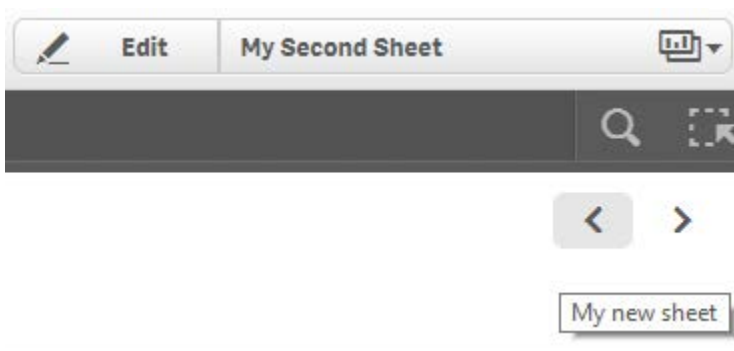
In the **Sheets** dialog, there are two ways of creating a new sheet. We can either click the **Create new sheet** button, or we can click on the + to the right of the existing sheets where it says **Create new sheet**.

We give the sheet a name and then we can open it and edit to add new objects, just as we did with the first sheet.

Navigating sheets

Once we have more than one sheet, we can navigate amongst them by selecting sheets from the **Sheets** dialog.

Also, if there is more than one sheet, in the top right-hand corner of the sheet we will see navigation buttons that allow us to move backward and forward through the sheets:



Using images and text

We can add objects to the sheet to display images and text. The text can be static or dynamic.

Understanding the content folder

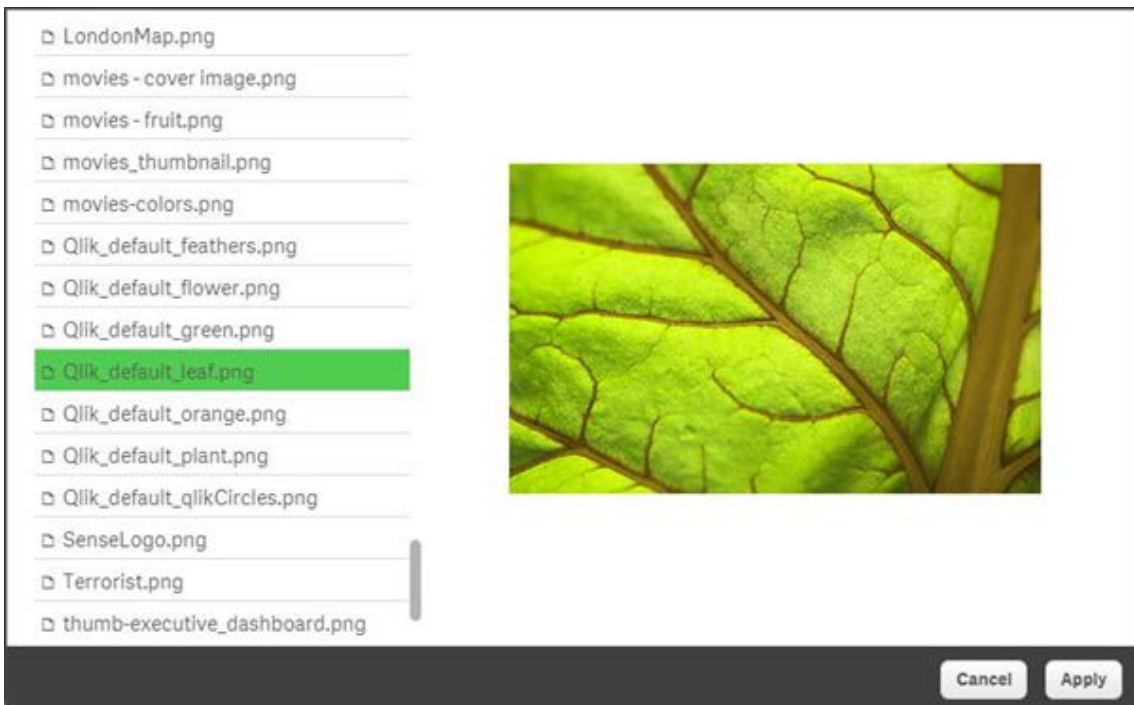
By default, all Qlik Sense Desktop content will be stored in **C: \Users\username\Documents\Qlik\Sense**. One of the sub-folders here is called **Content** and this has a sub-folder called **Default**. Any image files that are stored in this folder, or in any sub- folder under it, will be available for us to use in our application.



Have you saved recently?

Setting the app image

To the right of the app name on the toolbar there is a small **i** button which, when we hover over it, displays **Show app information**. When we click on that button we can edit the information for the app. When we click the **Edit** button we can change the name or edit the description. We can also click the camera icon to select an image for the application.

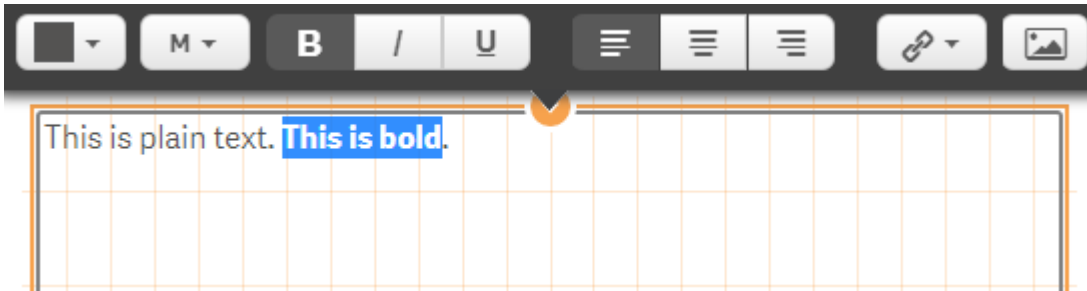


When we select an image for the application, this image will appear in the hub. We can click the **i** button again to dismiss the app information.

Adding a Text & Image object

The Text & Image object allows us to combine both text and images to a sheet.

If we edit a sheet and drag a Text & Image object on, we can then click into it and start typing plain text. The toolbar allows us to format the text with various options:



Telling stories

A new feature in Qlik Sense is the ability to take our analysis, annotate it with our own text and then share that with other Qlik Sense users. We take analyses and create stories.

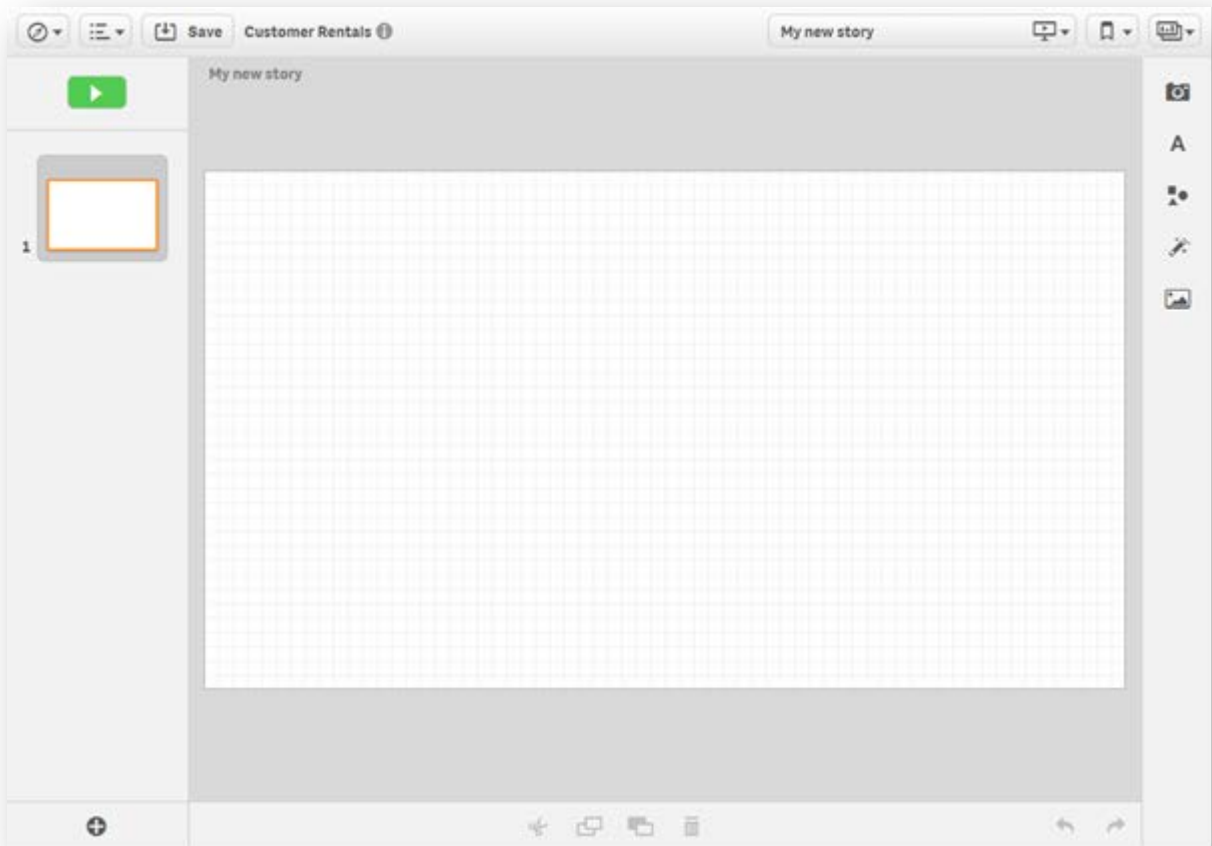
Taking snapshots

When we discover something in our data, we can take a snapshot of the chart. The snapshot stores an image of the chart as well as the set of selections that was active then the snapshot is taken.

To take a snapshot, we simply click on the camera icon on the toolbar and then click on the chart that we want to snapshot. We can take snapshots of as many charts as we want.

Creating a new story

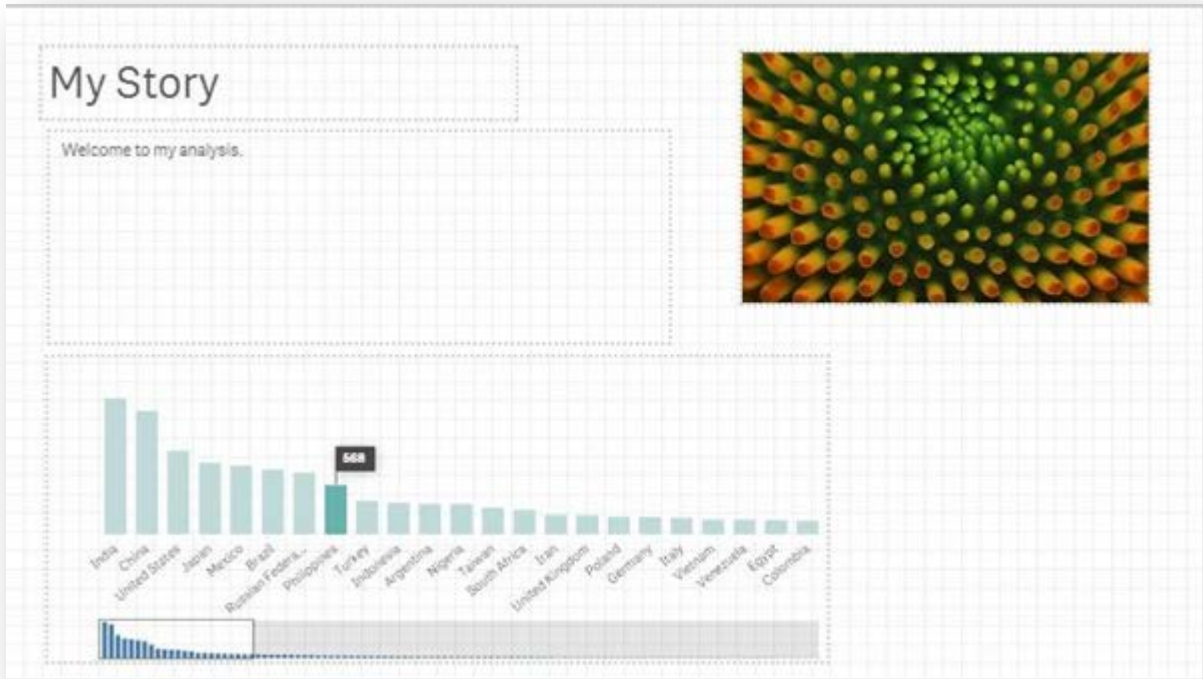
To create a new story we click the **Stories** button on the toolbar and click on **Create new story**. Give the story a name and then click on it to open. A new, blank page appears:



Adding content to the story

There are five buttons on the right-hand side that allow us to add new content:

- Snapshot library – This gives us access to all the snapshots that we have taken.
- Text library – We can add titles and paragraphs to our story.
- Shape library – There are multiple different shapes that we can add to highlight items in the story.
- Effect library – We can add effects such as automatically highlighting values.
- Media library – Images from our content folder are available to add to the story. We can add multiple items to the page and position as we want:



Adding new slides

By clicking the Add slide button on the bottom left-hand corner, we can add three different kinds of slide:

- Blank – a blank slide is just as when we first created the story. We can start adding our own content.
- Sheet – A sheet slide allows us to include a whole sheet from the application. This option has the sheet fill the whole slide.
- Sheet left aligned – With this option, the sheet is left aligned which leaves room for us to add additional content on the right-hand side.



Playing the story

To play the story we click the **Play the story** button on the top-left of the screen.

The play mode is a little like a slideshow in PowerPoint. However, the slides here are actually interactive.

Where there is a sheet object on the slide, we can click onto it and start interacting with it in the same way as we do with the app.

When there is a snapshot chart on the slide, we can right-click on that snapshot and select **Go to source** – it will open the application on the sheet that the snapshot came from with the selections that were there when the snapshot was taken.

Sharing apps with other users

Qlik Sense Desktop is a free product from Qlik. We can use it freely at home or in our business and we can share content that we create with other users that also have Qlik Sense Desktop.

Sharing files

All Qlik Sense Desktop files are stored in C:\Users\username\Documents\Qlik\Sense\Apps. Locate the file, with the qvf extension, that you want to share and simply email it, put it on a USB key or use a file share service like Dropbox to share it with others.

Once a user copies the qvf file into their own Documents\Qlik\Sense\Apps folder and it will then appear in their hub the next time that they open Qlik Sense Desktop.

Note that if you have custom images in your Documents\Qlik\Sense\Content\Default folder then you will also need to share these with other users.



Uploading to Qlik Cloud

Qlik Cloud is a service that Qlik have established to allow users to share Qlik Sense applications with up to three other users.

Log onto Qlik Cloud at <https://www.qlikcloud.com/login>. You can then upload your qvf files and share with other users.

Upload the Qlik Sense Server

If your organization has a Qlik Sense Server, qvf files can be uploaded to the server via the Qlik Management Console (QMC). Once uploaded, the app can be published to other users and will appear on their hub.

Summary

In this chapter we have had an introduction to using Qlik Sense Desktop to create analyses from the data file that we prepared in Pentaho Kettle. We learned how easy it is to simply drag a data file onto a new application to load the data. It is equally simple to create new charts and other objects to allow us to analyse our information.

We learned about how we can share the results that we discover by using the story telling feature. We also saw how we can easily share our Qlik Sense Desktop apps with other users.

For more information on getting started with Qlik Sense Desktop, see: <http://www.qlik.com/us/explore/products/sense/desktop/get-started>

For more information on getting started with Pentaho Kettle, see: <http://wiki.pentaho.com/display/EAI/Getting+Started>



APPENDIX



Appendix

In the Appendix we will cover off some items that are not correct to cover in the mainline of the book. These are the topics we'll be covering in this chapter:

- Installing the pagila database in PostgreSQL
- Connecting to the pagila database from Kettle



Installing the pagila database in PostgreSQL

This section is entirely optional as the tables from the database have already been extracted to the csv and made available to you in the download. It is provided here for those who may be interested.

The pagila database is a port of the Sakila sample database from MySQL which was based on an original Dell project to implement a DVD rental store. There are 15 tables (actually 21, but 6 are actually splits of one of the others). It includes tables such as customer, store, film, etc.

Installing PostgreSQL

If you need to install PostgreSQL, it is available from <http://www.postgresql.org/download/windows/>

Execute the downloaded file and follow the steps. Take careful note of the password that you set for the postgres user.

Downloading and extracting pagila

The pagila database (pagila-0.10.1.zip) should be downloaded from http://pgfoundry.org/frs/?group_id=1000150&release_id=998

Unzip the downloaded file to an appropriate location, for example, C:\Temp\pagila-0.10.1

Once unzipped we can run a **Command Prompt** and execute the following commands.



To create a new database called "pagila":

```
"C:\Program Files\PostgreSQL\9.4\bin\psql" -U postgres -c "CREATE DATABASE pagila"
```

To configure the database schema:

```
"C:\Program Files\PostgreSQL\9.4\bin\psql" -U postgres -d pagila -f
```

```
"C:\Temp\pagila-0.10.1\pagila-schema.sql"
```

To restore the database from the sql file:

```
"C:\Program Files\PostgreSQL\9.4\bin\psql" -U postgres pagila < "C:\Temp\pagila-0.10.1\pagila-data.sql"
```

Finally, run the ANALYZE command to update the indexes, etc.:

```
"C:\Program Files\PostgreSQL\9.4\bin\psql" -U postgres -d pagila -c "ANALYZE"
```

Creating the csv files from pagila

After you have successfully installed the pagila database, you can try opening the file, Extract

Files from pagila to csv.ktr, in Kettle to see how the csv files were created.

Connecting to the pagila database from Kettle

These instructions show us how to connect to the pagila database in PostgreSQL. The steps here are very similar for connecting to most enterprise databases.

Creating the database connection

First, open a transformation or add a new one. We can't create a database connection without having either a transformation or a job open.

From the **Tools** menu, select **Wizard | Create database connection....** The **Select database name and type** wizard appears.



Name of the database connection	pagila
Type of database to connect to	<ul style="list-style-type: none">HypersonicIBM DB2ImpalaInfobrightInformixIngresIngres VectorWiseIntersystems CacheKettle thin JDBC driverKingbaseESLucidDBMaxDB (SAP DB)MonetDBMS AccessMS SQL ServerMS SQL Server (Native)MySQLNative MondrianNeoviewNetezzaOpenERP ServerOracleOracle RDBPalo MOLAP ServerPostgreSQLRedshiftRemedy Action Request SystemSAP ERP SystemSQLiteSybaseSybaseIQTeradata
Type of database access to use	<ul style="list-style-type: none">Native (JDBC)ODBCJNDI

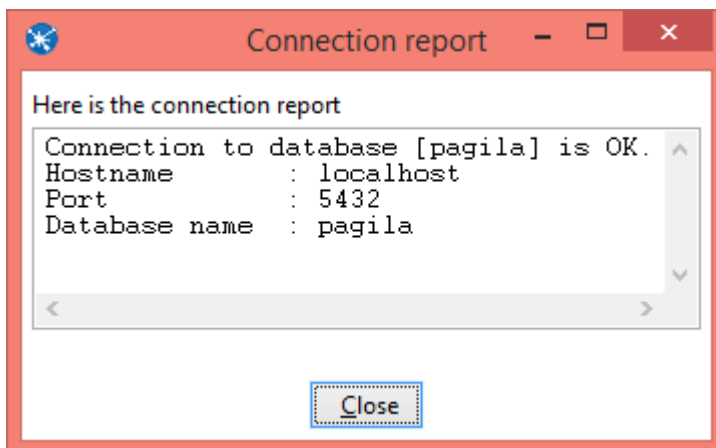


Enter **pagila** for the name and select **PostgreSQL** for the type. Usually, the **Native (JDBC)** option is the correct one for database access, but you can use ODBC or JNDI if available and you have the relevant drivers available and configured. Click the **Next** button.

In the settings dialog, enter the correct settings for the database. For the pagila database on PostgreSQL running on your local PC, the settings will look like this:

Host name of the database server	<input type="text" value="localhost"/>
The TCP/IP port	<input type="text" value="5432"/>
The name of the database	<input type="text" value="pagila"/>

After entering the correct settings, click **Next**. In the **Set the username and password** dialog, enter the correct username (postgres is the default user for PostgreSQL) and password (the one that you configured when installing PostgreSQL). Click the **Test database connection** button. You should see a dialog like this:

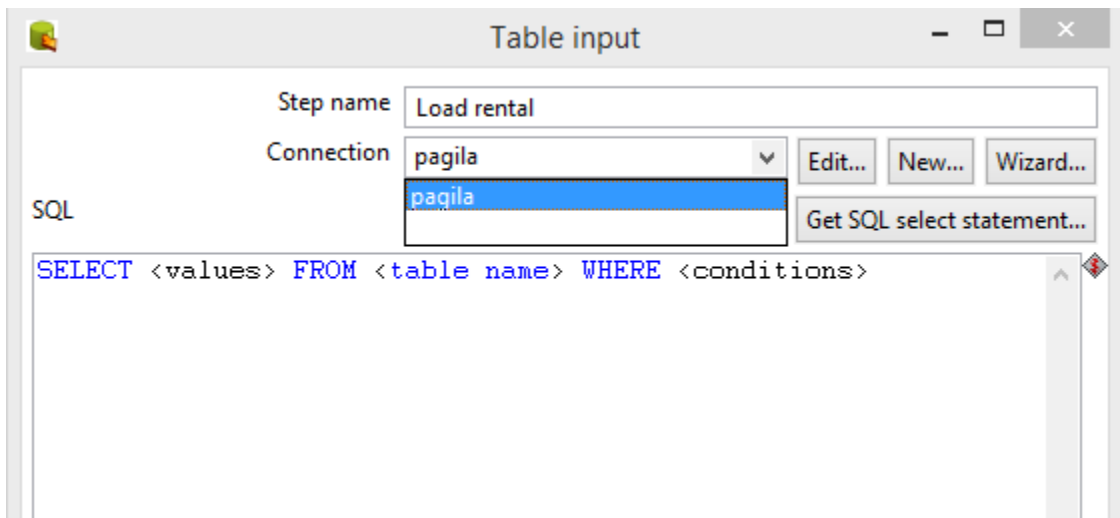




Click the **Close** button and then click the **Finish** button.

Adding a Table Input

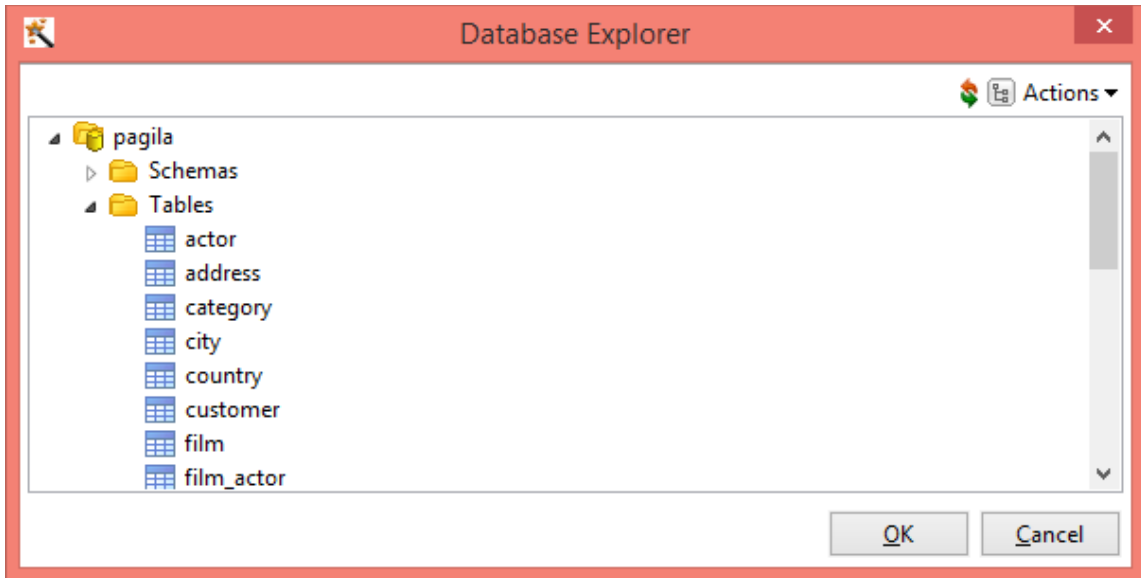
Drag a **Table Input** from the list of available inputs in the **Design** tab onto the transformation. Double-click to set the properties:



Note that the pagila connection that we configured in the last section will be listed by default. If we had not added the database connection then, we have another opportunity to add it here by clicking the **New** or the **Wizard** button.



Click the **Get SQL select statement...** button. A dialog opens to allow us to select a table:



Open out the list of tables and select **rentals** then click **OK**. We will be prompted if we want to list the fields or not; it is often better to select **Yes** as we can then easily modify the list of fields coming from the database.

Click **OK** to close the **Table Input** dialog. Now we carry on exactly the same as loading from a file.



ABOUT CAPVENTIS

Capventis is a specialist business & technology consultancy company based in the UK and Ireland, focusing on the areas of Business Intelligence (BI), Customer Relationship Management (CRM) and Customer Experience (CX). CapricornVentis Ltd (Capventis) was formed in 2006 with the merger of two companies, founded in 1996. From roots in Ireland and the Netherlands, Capventis now operates from two bases in Ireland and the UK.

Our Mission is to enable exceptional, sustainable value for all our stakeholders: customers, employees, partners and investors by delivering *brilliant* solutions for Business Intelligence (BI), Customer Relationship Management (CRM) and Customer Experience (CX), using our specialist knowledge and experience. By *brilliant*, we mean fast, innovative, relevant and successful.

We have three Practice lines:

Business Intelligence (BI) – managing information challenges, from simple delivery of visual dashboards to more in-depth and complex, analytics, decisions support and insight , data discovery, decision support and complex analytics and insight.

CRM/CX

- **Customer Relationship Management (CRM)** – ensuring that internal resources are structured, aligned and equipped to deliver maximum customer value
- **Customer Experience Management (CX)** – capturing and understanding how customers perceive and feel - the company, its brand, products, people and services, etc., - and taking action to improve on and exceed expectations.

Capventis Advisory – leveraging our executive-level relationships, approach, best practice and experience to define and align winning strategies and solutions for BI, CRM and CX.

Our value proposition is shaped by;

People. We are very specialised and skilled with a focus on working side by side with our clients team, making best use of our joint capability. We believe in learning and developing together, and promote knowledge transfer, to drive up mutual value and success.

Technology. We look for, and partner with the best technology vendors, platforms and solutions. We believe it critical to understand the technology potential when blueprinting the solution, but the business always comes first.

Approach. Our Blueprint methodology – developed with clients – is based on principles of client outcome and value. We seek to manage and deliver solutions that are aligned, relevant, and easy to execute. Brilliant-Basics is about establishing early value for each project, followed later by a process of evolution and continuous improvement.

Experience. Our team is very experienced in terms of subject area, analysis & design, project management and industry specialism.

We have clients across a range of industries including Financial Services, Consumer Goods, Retail, Energy & Utility, Manufacturing & Distribution, Healthcare & Life Science, Property, Public Sector and Local Government, and in the UK, Ireland, Mainland Europe, Asia, USA and Israel.



ABOUT THE AUTHOR

Stephen Redmond is the CTO and Qlik Luminary at Capventis (www.capventis.com), a QlikView Elite Partner.

He is the author of several books, including *Mastering QlikView*, *QlikView for Developers Cookbook* and *QlikView Server and Publisher* - all three published by Packt Publishing (www.packtpub.com). He is also the author of the popular DevLogix series for SalesLogix developers.

In 2006, after many years of working with CRM systems, reporting and analysis solutions, and data integration, Stephen started working with QlikView. Since then, CapricornVentis has become QlikView's top partner in the UK and Ireland territories, and with Stephen as the head of the team, they have implemented QlikView in a wide variety of enterprise and large-business customers across a wide range of sectors, from public sector to financial services to large retailers.

In 2014, Stephen was awarded the Luminary status by Qlik in recognition of his product advocacy. This status was re-awarded in 2015.

He regularly contributes to online forums, including the Qlik Community. His QlikView blog is at www.qliktips.com, and you can follow him on Twitter at @stephencredmond where he tweets about QlikView, BI, data visualization, technology in general, and occasionally, on marathon running.



COPYRIGHT

© 2015, CapricornVentis Limited, Block AD, Cherrywood Business Park, Loughlinstown, Dublin 18, IRELAND.

All rights reserved. This document is protected by copyright and is distributed under licenses restricting its use, copying and distribution. No part of this document may be reproduced in any form by any means without prior written authorization of Capventis.

First Edition April 2015

TRADEMARKS

QlikView and Qlik Sense are registered trademarks of Qlik.

Pentaho Data Integration and Kettle are registered trademarks of Pentaho, Inc. PostgreSQL is a registered trademark of The PostgreSQL Global Development Group.

Other product names may be trademarks or registered trademarks of their respective companies and are hereby acknowledged.

DISCLAIMER

The author has thoroughly reviewed this work. All statements, technical information, and recommendations in this manual and in any guides or related documents are believed reliable, but the accuracy and completeness thereof are not guaranteed or warranted, and they are not intended to be, nor should they be understood to be, representations or warranties concerning the products described. The author assumes no responsibility or liability for errors or inaccuracies with respect to this publication or usage of information. Further, the author reserves the right to make changes to the information described in this manual at any time without notice and without obligation to notify any person of such changes.

Published by Capventis

Block AD, Cherrywood Business Park, Dublin 18Ireland

CityPoint, 1 Ropemaker Street, London EC2Y 9HT, United Kingdom

www.capventis.com

Knowledge Centre

The Capventis knowledge centre is a library of useful documents to educate and inform on several aspects of business intelligence, Customer Relationship Management and Customer Experience Management. You have free, immediate and unlimited access to our ever-expanding library.

[Gain Access to Multiple Case Studies, Whitepapers, Industry Articles and Reports here.](#)