



London Air Quality Demo

Handbook

Table of Contents

Introduction	3
Problem Statement.....	3
Brief	3
Important aspects about jobs and transformations.....	4
Important aspects for the exercises.....	5
Creating the transformation.....	6
Gathering and loading data	6
Get started.....	6
Use transformation parameter to call site info API to get current list of monitoring sites.....	7
Make API calls to each monitoring site and check for HTTP response code.....	11
Read in returned data in JSON format and discard values	16
Lookup postal code from text file.....	18
Create fields for reporting and select required data to load into table	22
Dimensional tables	27
Months table	27
Days table	29
Creating Visualizations	32
Important aspects of the Data Source Wizard.....	32
Important aspects of Analyzer	33
Important aspects of Dashboard Designer	35
Get started.....	37
Creating the data source.....	37
Creating the Average Pollutant by Hour Analysis.....	51
Creating the Geospatial Report.....	59
Creating the Air Quality Dashboard	63

Introduction

Problem Statement

Air pollution causes 40,000 early deaths a year in the UK¹. 9,000 of those affected are residents of London. The health problems resulting from exposure to air pollution also have a high cost to society and business, our health services, and people who suffer from illness and premature death. In the UK, these costs add up to more than £20 billion every year¹. In addition to these costs the EU Commission has enacted legislation that cities in the EU must abide by or risk being fined².

Brief

In this workshop, we will use publicly available open data and the Pentaho platform to build a demonstration using data published by Kings College London on air quality in London to analyse pollution in the capital.

The data provided by King's College London includes hourly readings from sensor sites across London.

This workshop will make use of a breadth of Pentaho capabilities: onboarding multiple data sources, data cleansing, data blending and dashboards. This will enable us to report on air quality trends and lay the foundations for a data science team to use machine learning to predict air quality and breaches of thresholds.

What is it?

This handbook discusses how we can use data from the King's College London to do analytics on Air Quality. Air pollution is an increasing health issue in all major cities of the world. Air quality is well reported and cities are faced with legal action if they do not act to reduce levels of pollutants. To understand the problem, we turn to this well reported data. In this case we are using open data provided by King's College London who provide open data on the London Air website³.

During the exercises, we will build a transformation the fetch the London data. After the creation of the transformation we will build some analytics and a dashboard.

¹ <https://www.rcplondon.ac.uk/projects/outputs/every-breath-we-take-lifelong-impact-air-pollution>

² http://ec.europa.eu/environment/air/quality/legislation/existing_leg.htm

³ <https://www.londonair.org.uk/LondonAir/Default.aspx>

Important aspects about jobs and transformations

A **transformation** is a network of logical tasks called *steps*. Transformations are essentially *data flows*. The transformation is a directed graph of a logical set of data transformation configurations.

The two main components associated with transformations are **steps** and **hops**.

Transformation steps are the building blocks of a data flow, for example a text file input or a table output. There are over 300 steps available in Pentaho Data Integration and they are grouped according to function; for example, input, output, scripting, and so on. Each step is designed to perform a specific task, such as reading data from a flat file, filtering rows, and logging to a database. Steps can be configured to perform the tasks you require.

Transformation hops are data pathways that connect steps together and allow schema metadata to pass from one step to another. Hops determine the flow of data *through* the steps not necessarily the sequence in which they run. When you run a transformation, *each step starts up in its own thread and pushes and passes data*.

A **job** is a workflow-like model for coordinating resources, execution, and dependencies of data integration activities. Jobs aggregate up individual pieces of functionality to implement an entire process.

Jobs are composed of job hops, job entries, and job settings.

Job hops behave differently when used in a job. Besides the execution order, a hop also specifies the condition on which the next job entry will be executed. You can specify the Evaluation mode by right clicking on the job hop. A job hop is just a flow of control. Hops link to job entries and, based on the results of the previous job entry, determine what happens next.

The Evaluation modes are:

- **Unconditional:** Specifies that the next job entry will be executed regardless of the result of the originating job entry
- **Follow when result is true:** Specifies that the next job entry will be executed only when the result of the originating job entry is true; this means a successful execution such as, file found, table found, without error, and so on
- **Follow when result is false:** Specifies that the next job entry will only be executed when the result of the originating job entry was false, meaning unsuccessful execution, file not found, table not found, error(s) occurred, and so on

Job entries are the individual configured pieces, the primary building blocks of a job. (In data transformations, these individual pieces are called steps.) Job entries can provide you with a wide range of functionality ranging from executing transformations to getting files from a Web server.

Job settings are the options that control the behavior of a job and the method of logging a job's actions.

Transformation and jobs files are xml files and transformation names have a .ktr extension and job names a .kjb extension.

Important aspects for the exercises

The Design tab in the Pentaho Data Integration client has a search bar to search for steps with a specific word in its name.

In case of not knowing where a step is for or what some fields in a step mean you can look up the manual for a step in two ways:

- Press the help button in the step, or
- Look up the step at the following links:
 - For jobs:
<https://help.pentaho.com/Documentation/7.1/0L0/0Y0/0L0>
 - For transformations:
<https://help.pentaho.com/Documentation/7.1/0L0/0Y0/0K0>

Files

Files for the exercises are in the /home/pentaho/londonair folder. Londonair application logfiles need to be written there as well.

Database drivers

The exercises are done on the default Postgres database for the Pentaho Repository. This is fine for this 'learning' exercise, but not for production. It slows down the performance for reading and writing saved Pentaho objects.

You can use your own database, but you need to make sure that the **database JDBC driver** is installed and accessible for Pentaho. For this you need to install the JDBC driver (.jar file) in a Pentaho folder, with the help of the **distribute-files script** that can be found in the pentaho/jdbc-distribution folder. If the correct driver is not there you will get an error when trying to connect to the data source.

Creating the transformation

Gathering and loading data

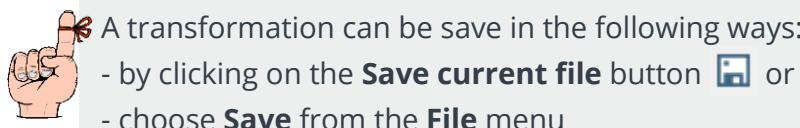
Pentaho Data Integration to aggregate, cleanse and blend data to drive analytics. To gather this data, we need to make API calls to each of the monitoring sites individually.

Get started

1. Open Pentaho Data Integration by clicking on the PDI icon 
2. Create a Pentaho Data Integration (PDI) transformation.

From the main menu choose **File > New > Transformation**

3.  Save your transformation before you proceed. It is good practice to prefix your transformation with **t_** i.e. **t_london_air_quality.ktr**



4. Right click on the canvas to specify some parameters we are going to use in the transformation.
5. In the Transformation properties window, we go to the **Parameters** tab and enter the following parameters:

Parameter	Default Value	Description
1 site_info_url	http://api.erg.kcl.ac.uk/AirQuality/Information/MonitoringSites/GroupName=London/json	Site Info API URL
2 base_aqd_url	http://api.erg.kcl.ac.uk/AirQuality/Data/Site/SiteCode=	Air Quality Site API URL
3 data_type	json	Data Type
4 start_date	StartDate=2017-09-15	Request from (inclusive)
5 end_date	EndDate=2017-09-18	Request to (exclusive)



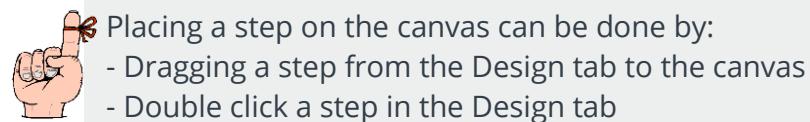
Parameter	Default Value	Description
site_info_url	http://api.erg.kcl.ac.uk/AirQuality/Information/MonitoringSites/GroupName=London/json	Site Info API URL
base_aqd_url	http://api.erg.kcl.ac.uk/AirQuality/Data/Site/SiteCode=	Air Quality Site API
data_type	json	Data Type
start_date	StartDate=2017-09-15	Request from (inclusive)
end_date	EndDate=2017-09-18	Request to (exclusive)

- a. Press Ok to save and close this window

Use transformation parameter to call site info API to get current list of monitoring sites

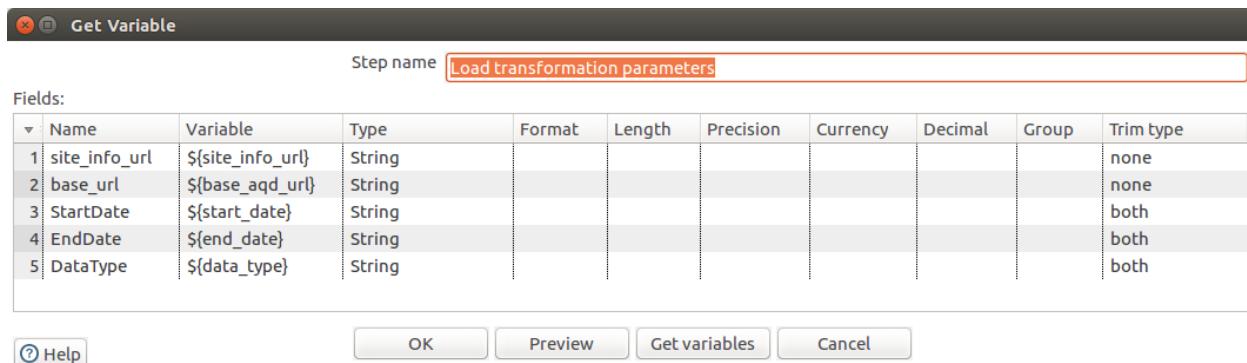
We first must make sure we have an up-to-date list of sites by making an initial call to the site info API.

6. Place the **Get Variables** step on the canvas to load transformation parameters



7. Double click on the **Get Variables** step to open specify the properties

- a. Give it a name and enter the following parameters



Name	Variable	Type
site_info_url	\${site_info_url}	String
base_url	\${base_aqd_url}	String
DataType	\${data_type}	String
StartDate	\${start_date}	String
EndDate	\${end_date}	String

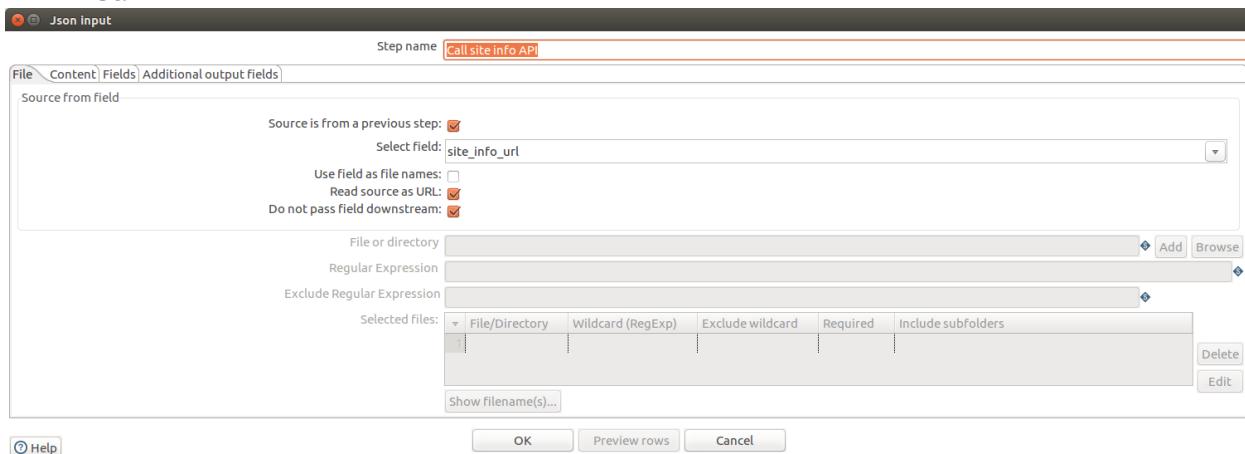
- b. Press OK to save and close this window
8. Drag the **Json Input** step on to the canvas
9. Create a hop between the **Get Variables** and **Json Input** steps

 To create a hop, click the source step, then press the <Shift> key down and draw a line to the target step. Alternatively, you can draw hops by hovering over a step until the hover menu appears. Drag the hop painter icon from the source step to the target step. Another way to create a hop automatically is by first selecting a step on the canvas and then instead of dragging a step from the Design tab to the canvas, <Ctrl>-double click the step to be added.
To split a hop, insert a new step into the hop between two steps by dragging the step over a hop. Confirm that you want to split the hop.

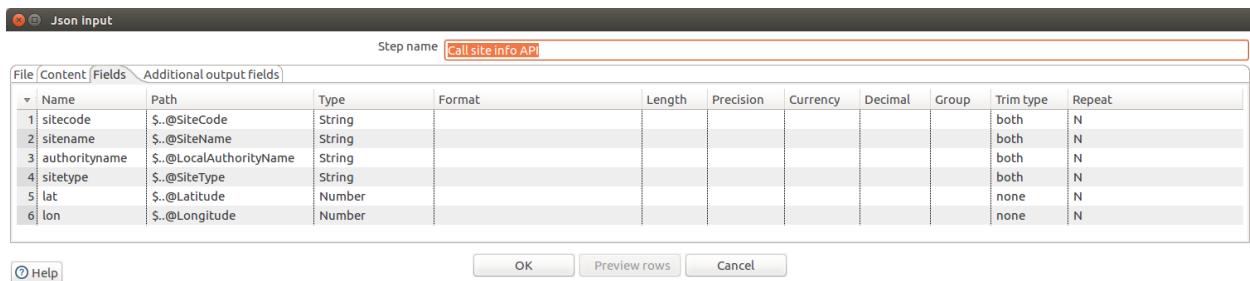
 The previous two actions can be handled in one. <Shift>-double click on a step in the Design tab will add the step, will create a hop and will open the added step.

10. Double click on the **Json Input** step to specify its properties

- a. Rename the **Step name** to Call site info API
- b. Enter the following configuration on the File tab. Select field contains a drop-down list.



- c. On the **Content** tab leave the fields as default
- d. On the **Fields** tab complete the table with the following information:



Help

OK Preview rows Cancel



Name	Path	Type
sitecode	\$..@SiteCode	String
sitename	\$..@SiteName	String
authorityname	\$..@LocalAuthorityName	String
sitetype	\$..@SiteType	String
lat	\$..@Latitude	Number
lon	\$..@Longitude	Number

- e. Press **OK** to save and close the window
- f. Run the transformation to check your progress and confirm the results on the **Preview data** tab of the **Execution Results**



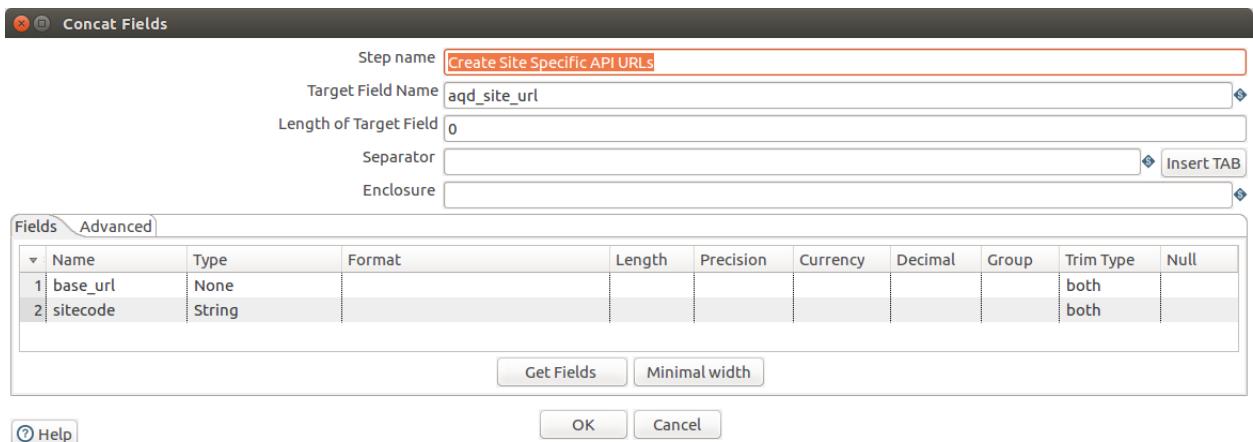
This step uses JSON dot notation using \$ to represent the root. For more information please see [the W3 website](#). Please also note that these are **case sensitive**.

11. Drag the a **Concat fields** to the canvas to create site specific API calls
12. Create a hop between the **JSON Input** and **Concat fields** step and choose **Main output of step** from popup



Only if the source step has a Main output and an error output for error handling a popup will appear to choose the proper hop.

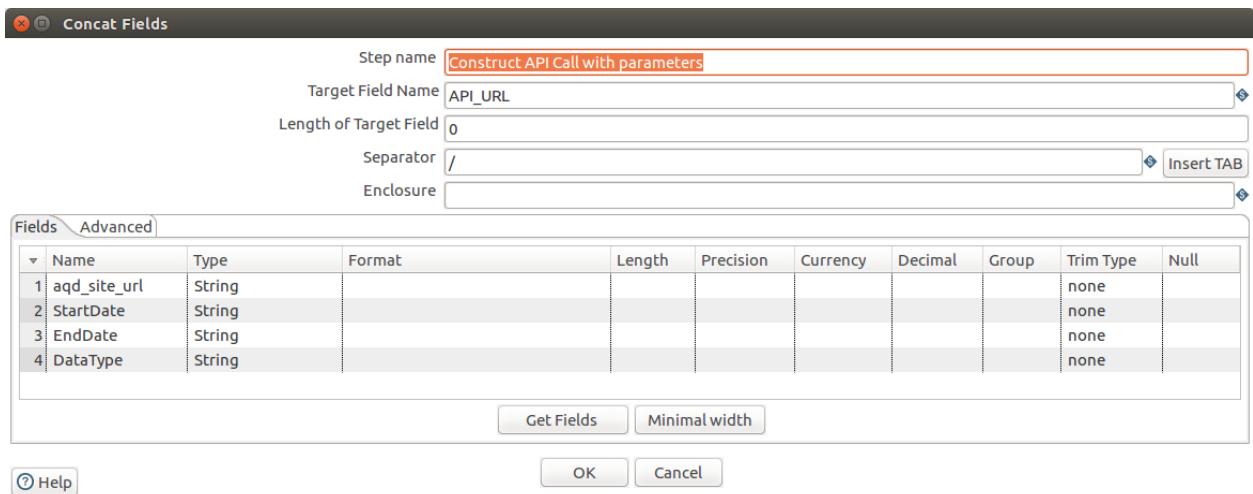
13. Double click on the **Concat Fields** step to specify its properties
 - a. Name: Create Site Specific API URLs
 - b. Enter the following configuration, using the Get Fields button to get the available fields in the transformation stream.



 Using the Get Fields button enters the available transformation stream files into the Fields table. You can delete unneeded fields by clicking on the line number and then delete them.

 A cell can be copied to all other rows by right clicking and selecting **Copy field value to all rows**.

- g. Press **OK** to save and close the window
14. Drag a second **Concat fields** to the canvas to add the dates and data type to the API URL
15. Create a hop between the first and second **Concat fields** steps
16. Double click on the second **Concat Fields** step to specify its properties
 - a. Name: Construct API Call with parameters
 - b. Enter the following configuration. Use the Get Fields button to get the available fields in the transformation stream.



- c. Press **OK** to save and close the window

17. Save the transformation

Your transformation should look *similar* to this:

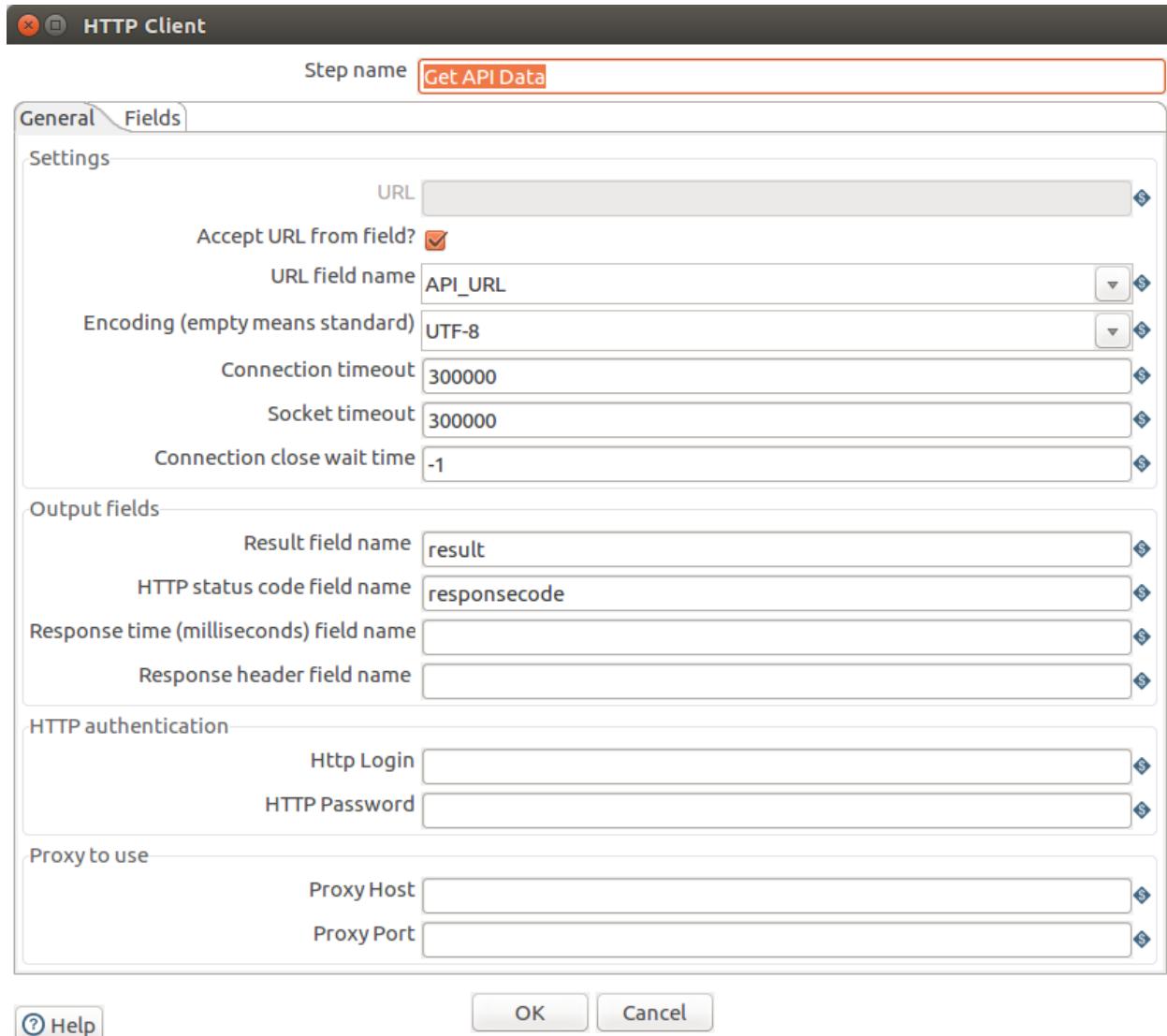


 If you have problems with leading and trailing white space, shown in the [Preview Data](#) tab of your [Execution Results](#), try changing the **Trim Type** to **both**.

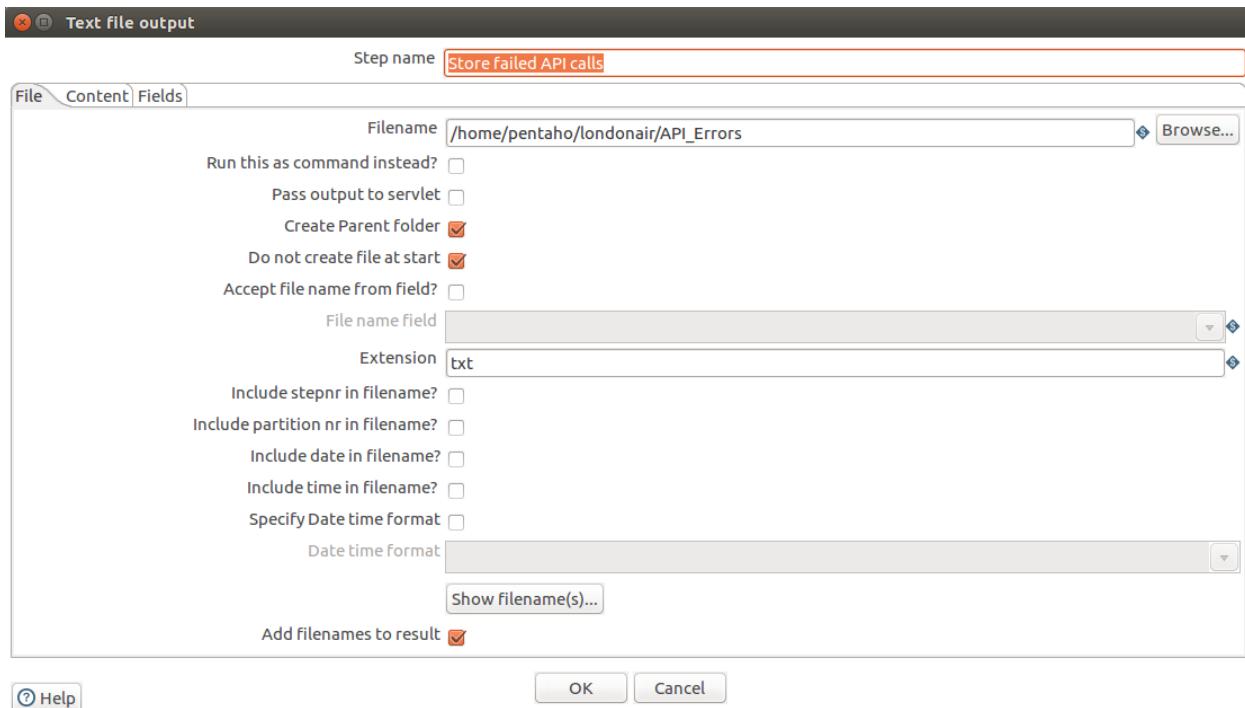
Make API calls to each monitoring site and check for HTTP response code

When making API calls we validate the request is being successful. If not we store the URL for later.

1. From the **Design** tab drag the **HTTP Client** step to the canvas to get API data
2. Create a hop between the **Concat Fields** and the **HTTP Client** steps
3. Double click on the **HTTP client** step to specify its properties
 - d. Name: Get API data
 - e. Enter the following configuration:



- f. Leave the Fields tab empty, meaning that in this case we do not have parameters and do not have a HTTP Header information.
 - g. Press **OK** to save and close the window.
4. Drag the **Text File Output** step to the canvas to store failed API calls
 5. Create a hop between the **HTTP Client** and the **Text File Output** steps and choose **Error handling of step** from popup
 6. Double click on the **Text File Output** step to specify its properties
 - a. Name: Store failed API calls
 - b. Enter the following configuration in the Files tab:



- c. Accept the defaults in the Content tab
- d. Enter the following configuration in the Fields tab. Use the Get Fields button.

Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim Type
1 base_url	String		85					none
2 DataType	String		85					both
3 StartDateString	String		85					none
4 EndDateString	String		85					none
5 StartDateStamp	Date							none
6 EndDateStamp	Date							none
7 StartDate	String		0					none
8 EndDate	String		0					none
9 sitocode	String							none
10 sitename	String							none
11 authorityname	String							none
12 sitetype	String							none
13 lat	Number							none
14 lon	Number							none
15 aqd_site_url	String		0					none
16 API_URL	String		0					none

- e. Press **OK** to save and close the window.

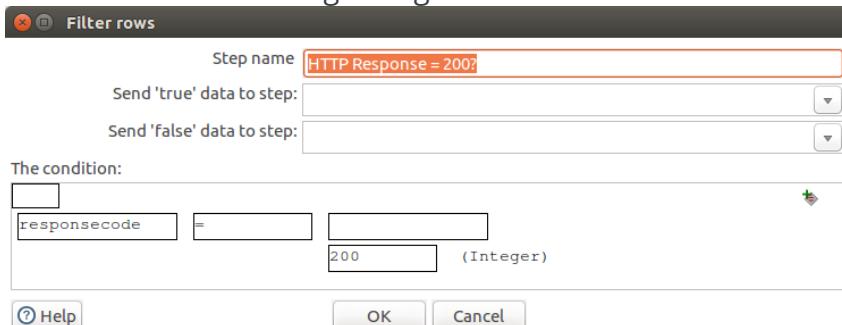
1. Drag the **Filter rows** step to the canvas to filter the rows with HTTP response code unequal to 200
2. Create a hop between the **HTTP Client** and the **Filter rows** steps and choose **Main output of step** from the popup

Now a warning window will popup. Click on the Copy button to make sure that all correct rows coming from the source step continue into the correct path of the flow (in this case the **Filter rows** step).

-  There are three choices:
- Distribute rows: destination steps will receive rows in turns
 - Copy rows: destination steps will receive all rows
 - Custom distribute rows: for load balancing

3. Double click on the **Filter rows** step to specify its properties

- a. Name: HTTP Response = 200?
- b. Enter the following configuration:

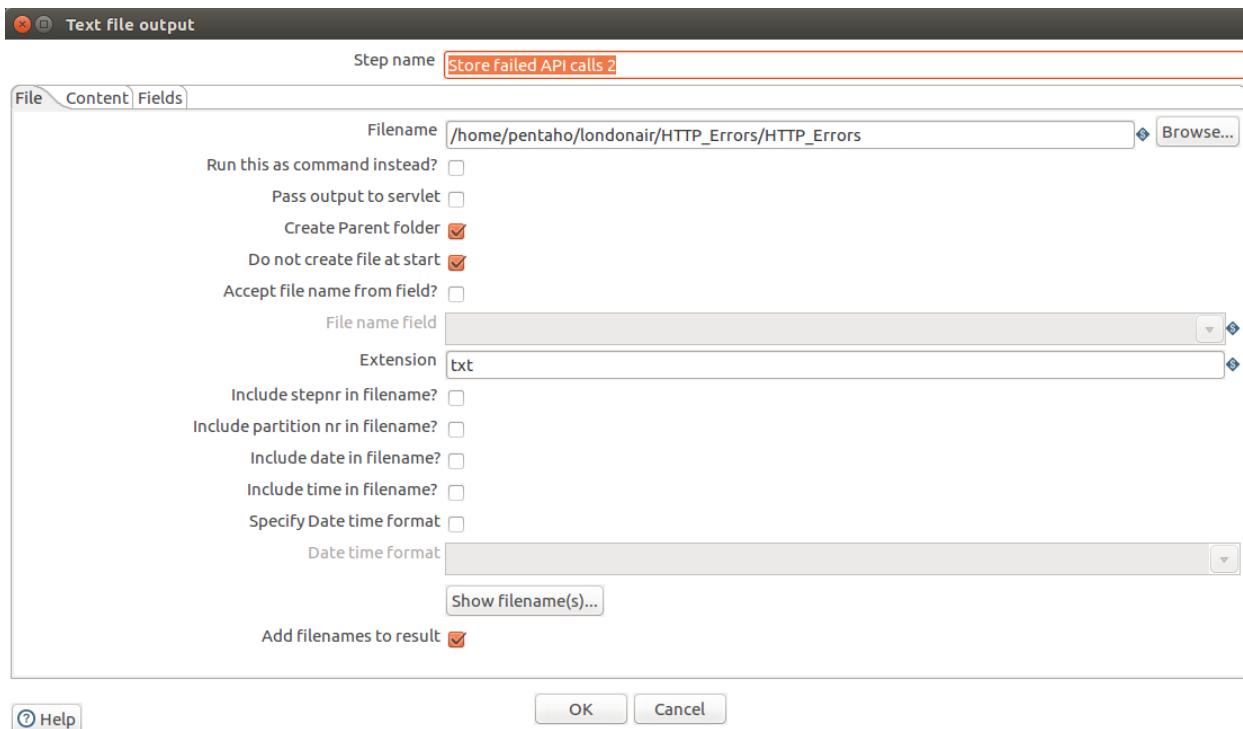


Please note that **the Send 'true' data step** and **Send 'false' data to step** fields cannot be filled in yet. These will be filled in automatically when we create the streams for true and false data.

- c. Press **Ok** to save and close the window
4. Drag the **Text File Output** step to the canvas to store failed API calls
5. Create a hop between the **Filter rows** and the **Text File Output** steps and choose **Result is false** from the popup

-  Notice the difference between the representation of the error handling flow and an error flow.

6. Double click on the **Text File Output** step to specify its properties
 - a. Name: Store failed API calls 2
 - b. Enter the following configuration:



- c. Accept the default configuration on the Content tab
- d. Enter the following configuration on the Fields tab:

Text file output

Step name **Store failed API calls 2**

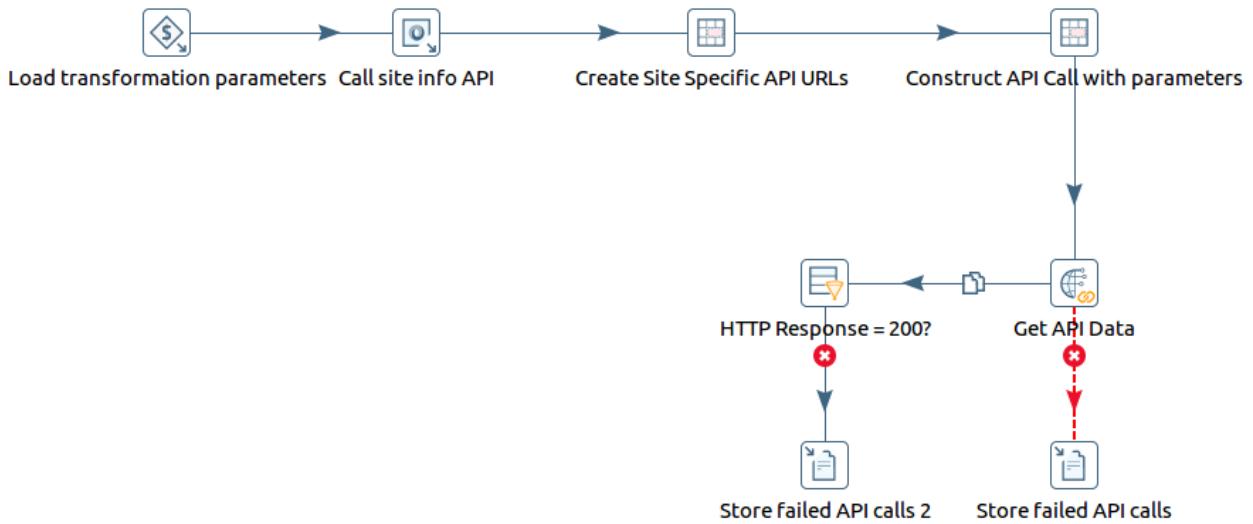
Fields

Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim Type	Null
1 base_url	String		85					none	
2 DataType	String		85					both	
3 StartDateString	String		85					none	
4 EndDateString	String		85					none	
5 StartDateStamp	Date							none	
6 EndDateStamp	Date							none	
7 StartDate	String		0					none	
8 EndDate	String		0					none	
9 sitecode	String							none	
10 sitename	String							none	
11 authorityname	String							none	
12 sitetype	String							none	
13 lat	Number							none	
14 lon	Number							none	
15 aqd_site_url	String		0					none	
16 API_URL	String		0					none	
17 result	String							none	
18 responsecode	Integer	##0;###0	0	.	,			none	

- e. Press **OK** to save and close the window

7. Save the transformation

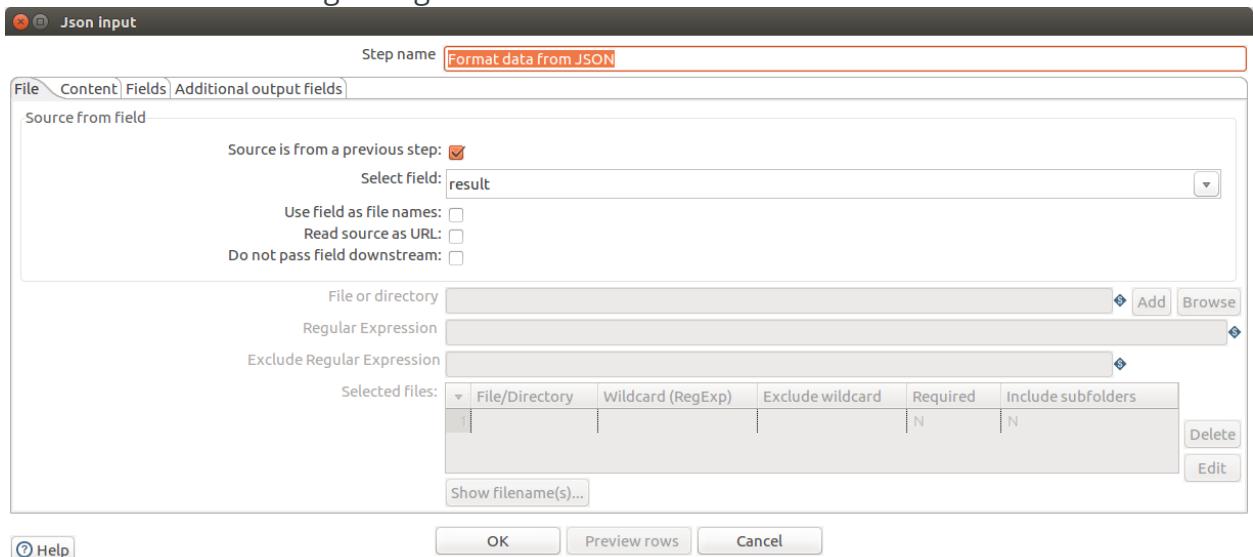
Your transformation should look *similar* to this:



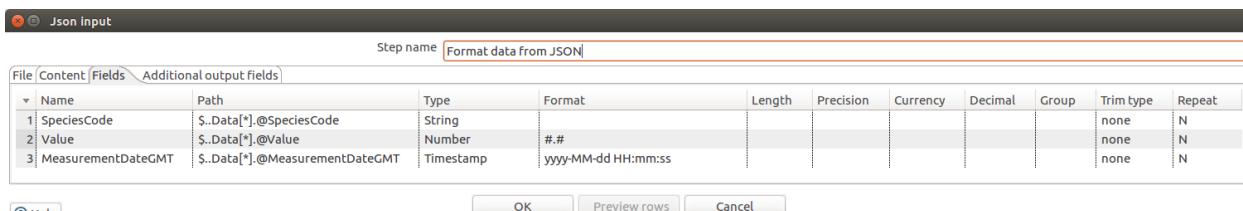
Read in returned data in JSON format and discard values

Once we extract the sensor readings from the JSON data we discard null values.

1. Drag the **JSON Input** step to the canvas to format data from JSON
2. Create a hop between the **Filter rows** and the **JSON Input** steps and choose the **Result is true** from the popup.
3. Double click on the **JSON Input** step to specify its properties
 - a. Name: Format data from JSON
 - b. Enter the following configuration on the File tab:



- c. Enter the following configuration on the Fields tab:

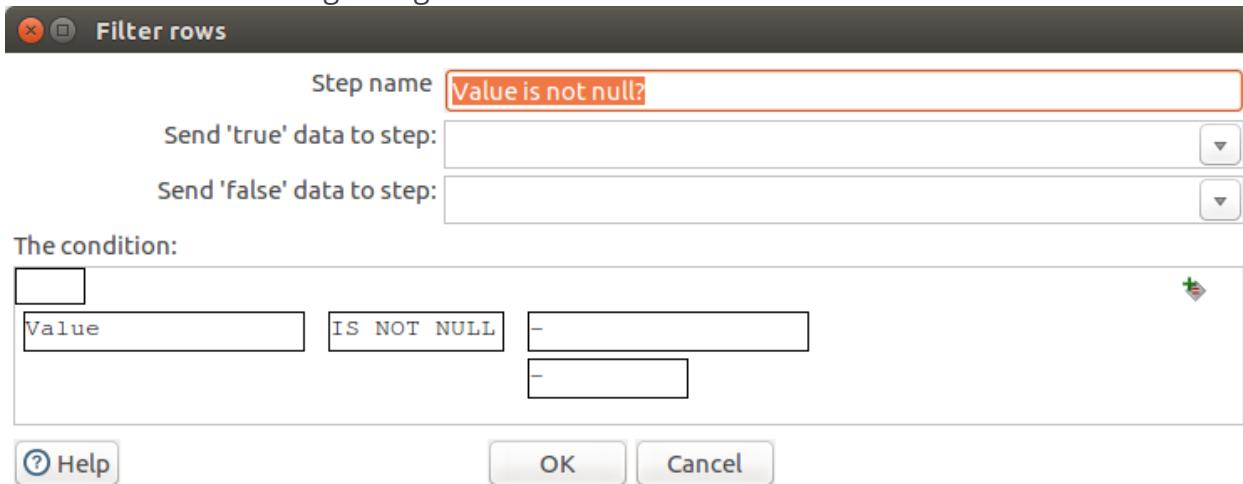


Name	Path	Type
SpeciesCode	\$.Data[*].@SpeciesCode	String
Value	\$.Data[*].@Value	Number
MeasurementDateGMT	\$.Data[*].@MeasurementDateGMT	Timestamp

- d. Accept the default configuration on the other tabs
- e. Press **OK** to save and close window

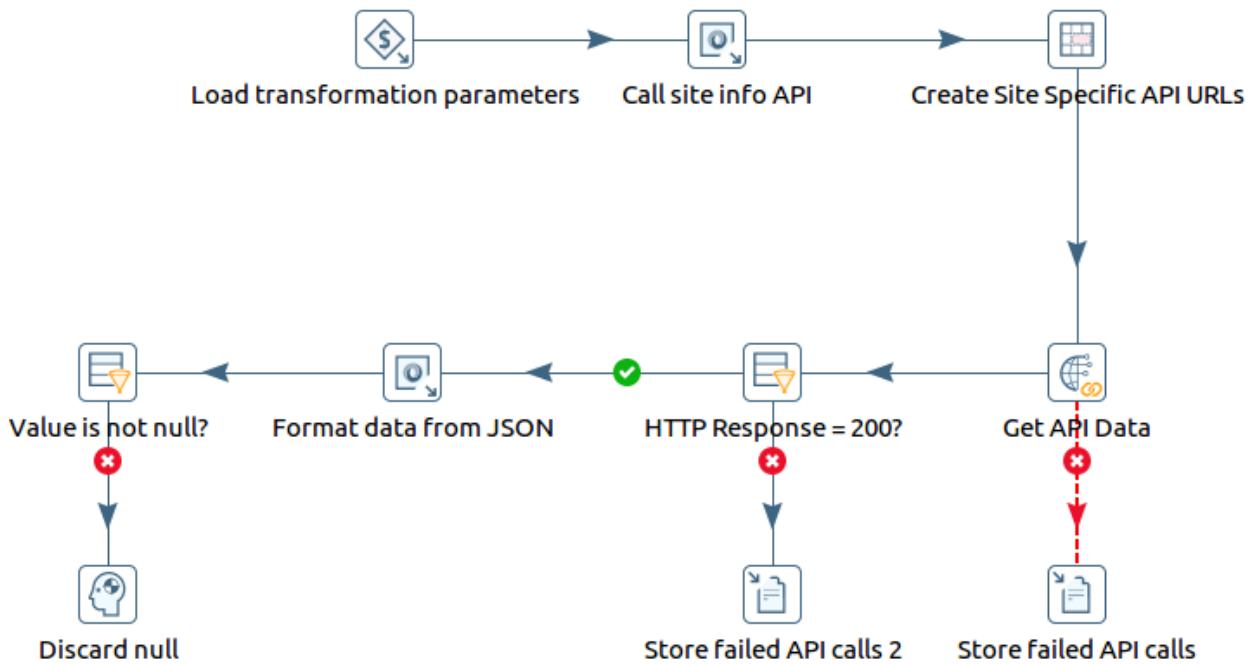
 If in your locality you use comma separators for decimals instead, you must force the decimal separator to a full stop in the **Decimal** column.

4. Drag the **Filter rows** step to the canvas to validate the value is not null
5. Create a hop between the **JSON Output** and **Filter rows** steps and choose **Main output of step** from the popup
6. Double click on the **Filter rows** step to specify its properties
 - a. Name: value is not null?
 - b. Enter the following configuration:



- c. Press OK to save and close this window
- 7. Drag the **Dummy (do nothing)** step to the canvas to discard null value
- 8. Create a hop between the **Filter rows** and **Dummy (do nothing)** steps and choose **Result is false** from the popup
- 9. Save the transformation

Your transformation should look *similar* to this:



Lookup postal code from text file

To add an additional dimension for reporting we lookup the postal code from a flat file and split it so we can report on a larger area.

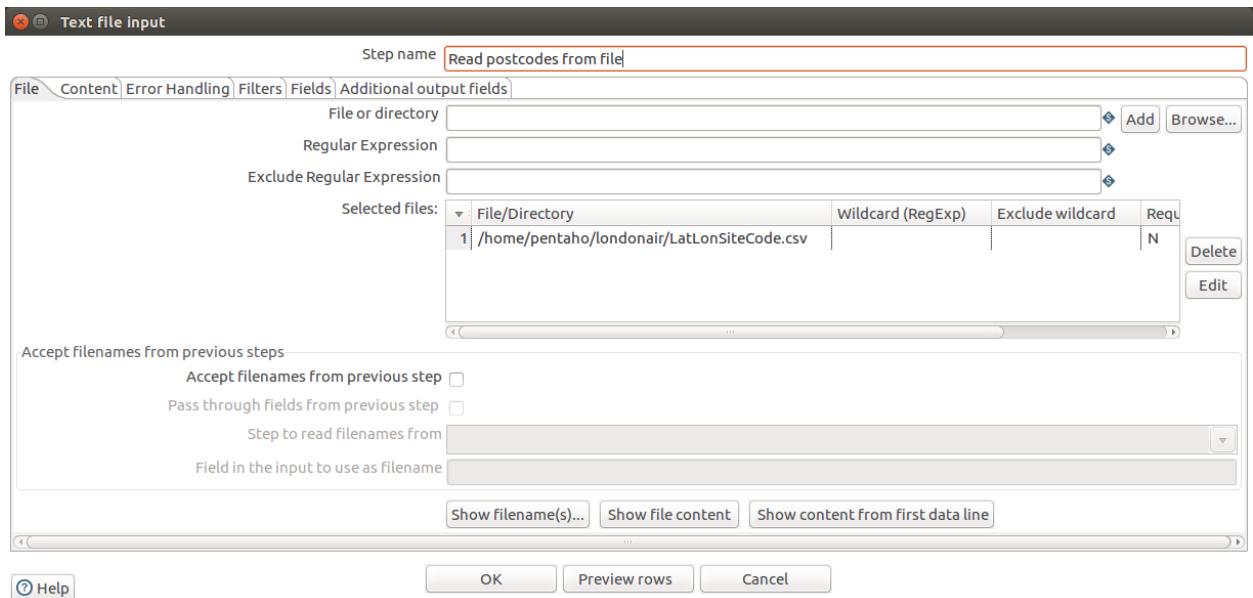
1. Drag the **Text File Input** step to the canvas to read postcodes from file



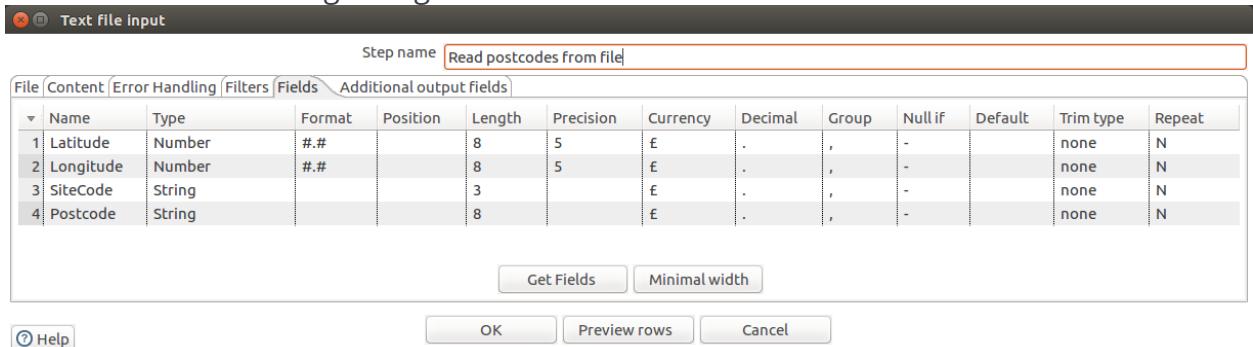
We are going to create a new input flow, hence place the File Input step higher than the existing flow. The current flow can be dragged down by drawing a box around all the steps and moving it down.

2. Double click on the **Text File Input** step to specify its properties

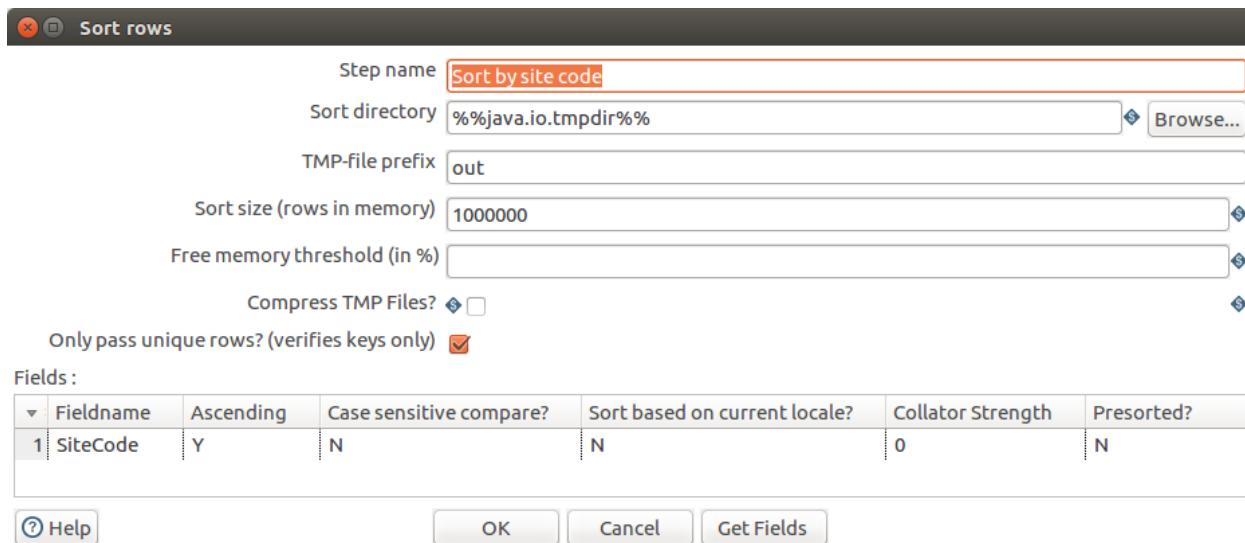
- a. Name: read postcodes from file
- b. Enter the following configuration on the File tab:



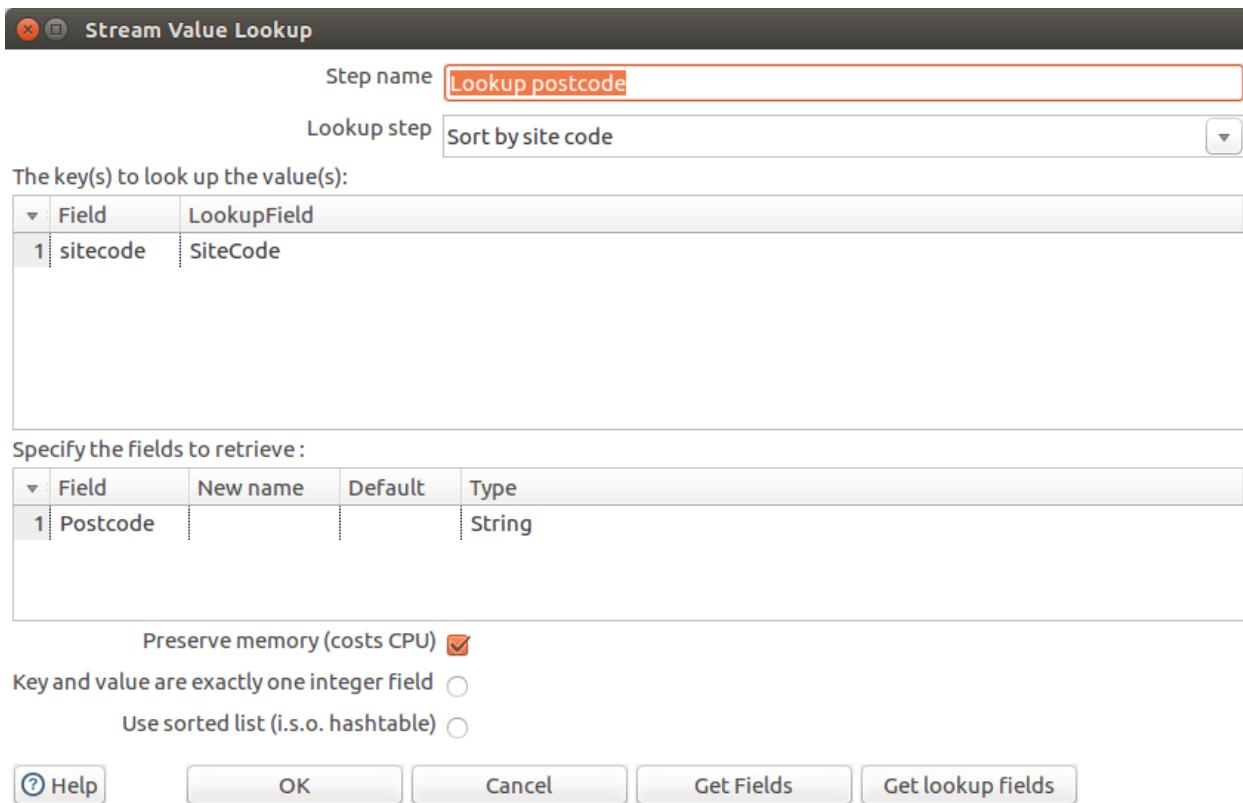
c. Enter the following configuration on the Fields tab:



- d. Accept the default configuration for the other tabs
 - e. Press **OK** to save and close the window
3. Drag the **Sort rows** step to the canvas to sort by site code
 4. Create a hop between the **Text File Input** and **Sort rows** steps
 5. Double click on the **Sort rows** step to specify its properties
 - a. Name: sort by site code
 - b. Enter the following configuration:

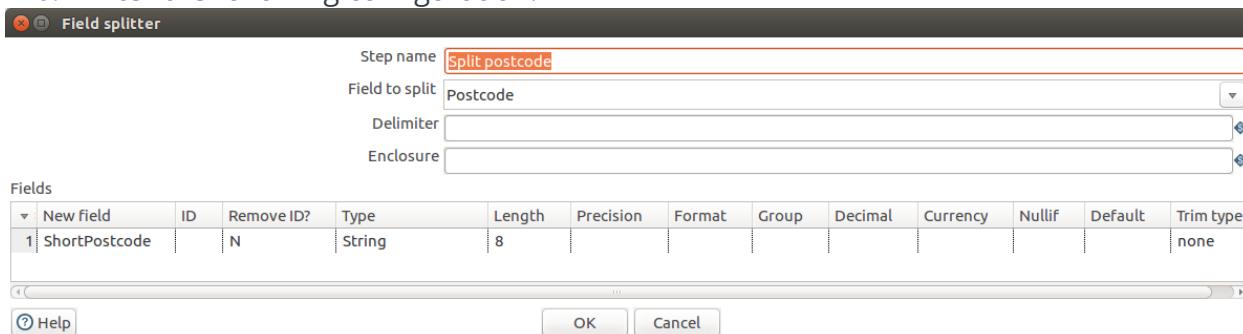


- c. Press **OK** to save and close the window
6. Drag the **Stream lookup** step to the canvas to lookup postcode
7. Create two hops:
 - a. From Sort rows to Stream Lookup, and
 - b. From Filter rows to Stream Lookup and choose Result is true from the popup
8. Double click on the **Stream Lookup** step to specify its properties
 - a. Name: Lookup postcode
 - b. Enter the following configuration, by first selecting the Lookup step:



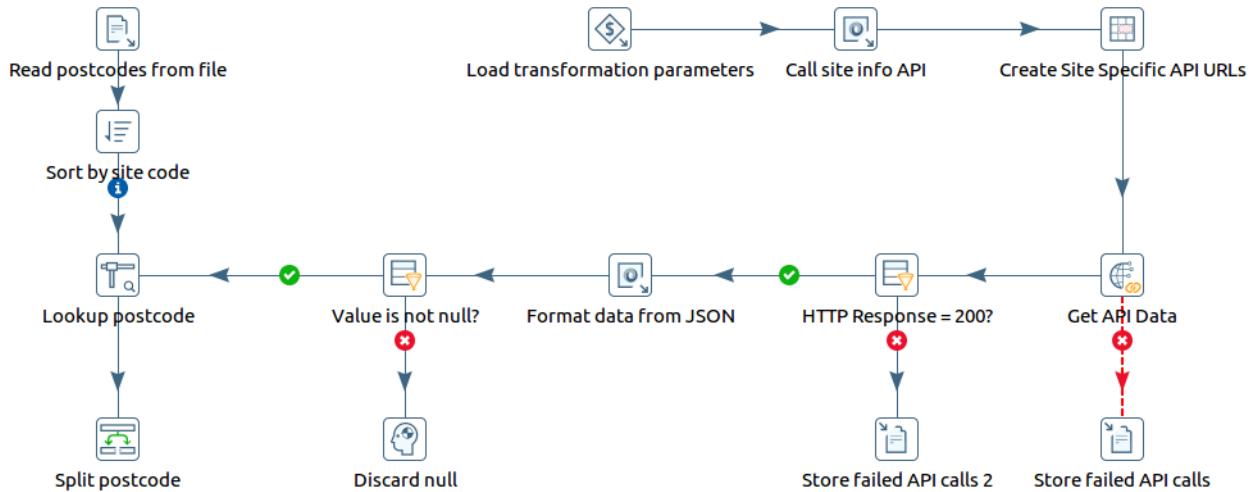
- c. Press **OK** to save and close the window
9. Drag the **Split Fields** step to the canvas to split postcode
10. Create a hop between the **Stream Lookup** and **Split Fields** steps
11. Double click on the **Split Fields** step to specify its properties

- a. Name: Split postcode
- b. Enter the following configuration:



- Note: Delimiter entry is a space
- c. Press **OK** to save and close the window
 12. Save the transformation

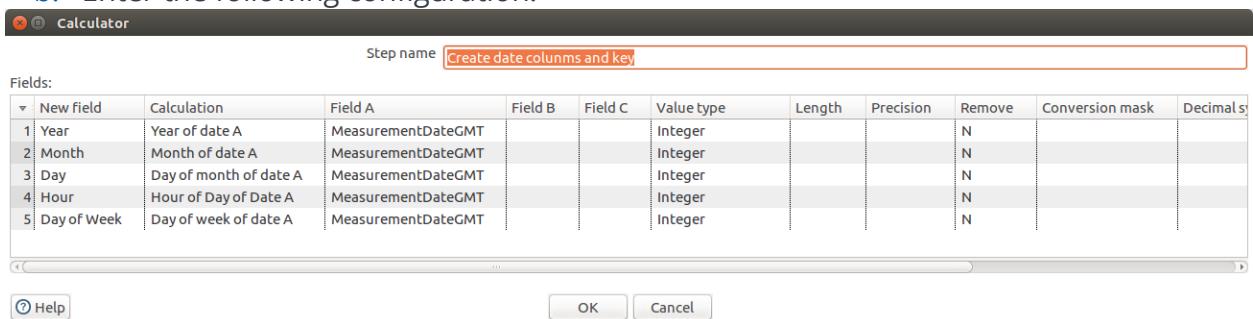
Your transformation should look *similar* to this now:



Create fields for reporting and select required data to load into table

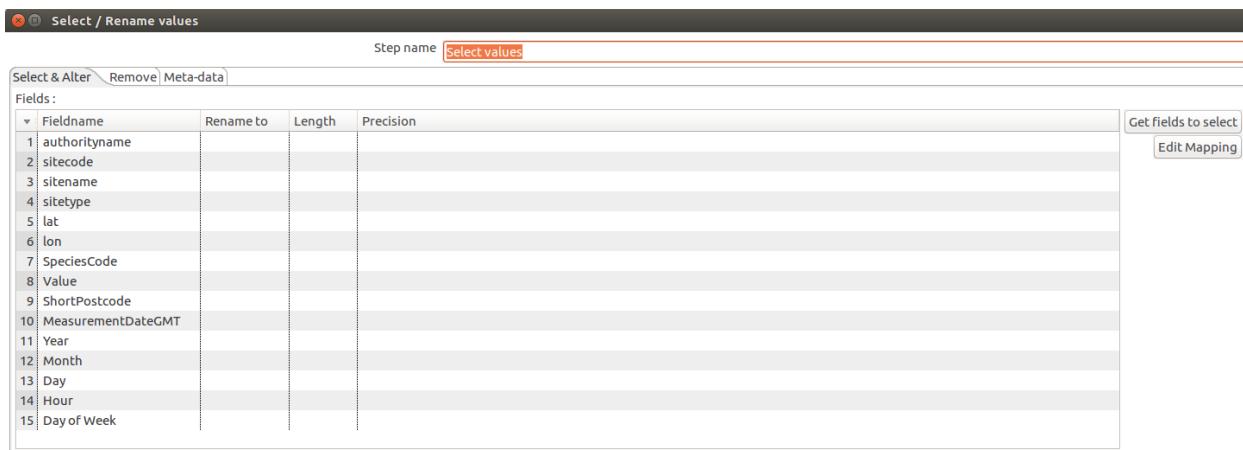
Finally, we use a calculator step to add time columns for reporting before loading the data our database

1. Drag the **Calculator** step to the canvas to create date columns and key
2. Create a hop between the Split Fields and Calculator steps
3. Double click on the **Calculator** step to specify its properties
 - a. Name: Create date columns and key
 - b. Enter the following configuration:

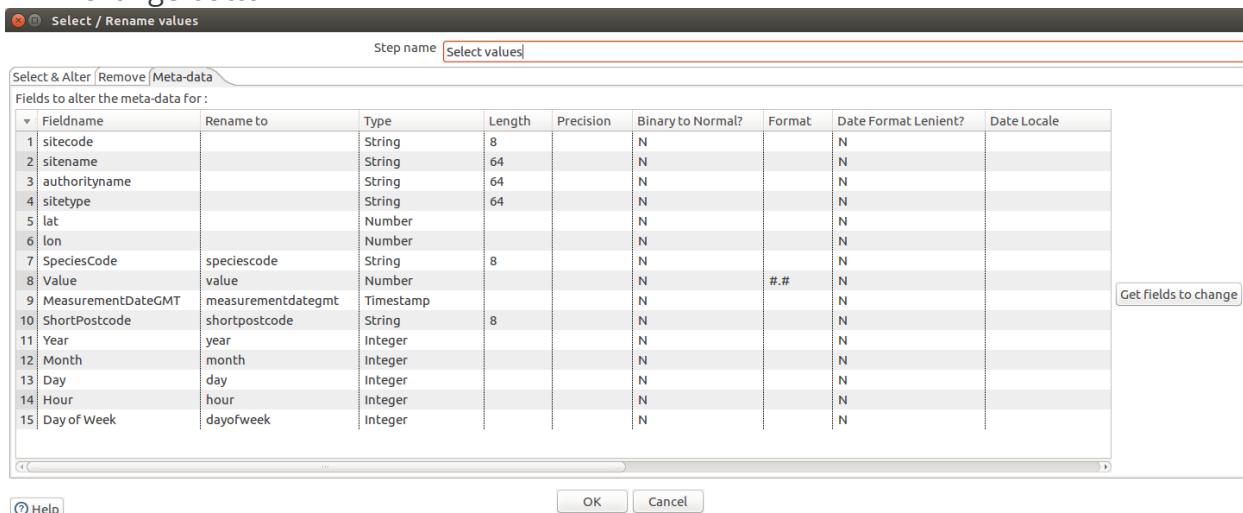


The functions in the calculation field can be selected for a drop-down list.

- c. Press OK to save and close this window
4. Drag the **Select values** step to the canvas to select the needed values
5. Double click on the **Select values** step to specify its properties
 - a. Name: select values
 - b. Enter the following configuration on the Select & Alter tab, using the Get fields to select button:

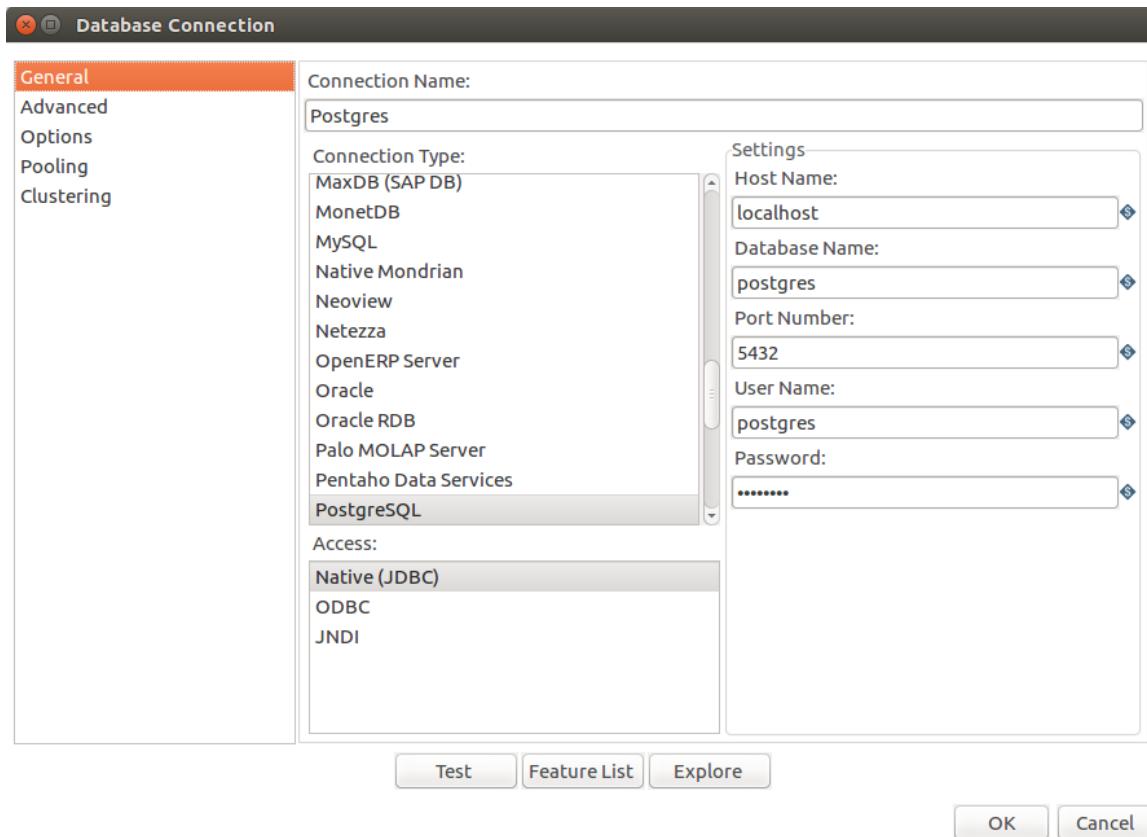


- c. Enter the following configuration on the Meta-data tab, using the Get fields to change button



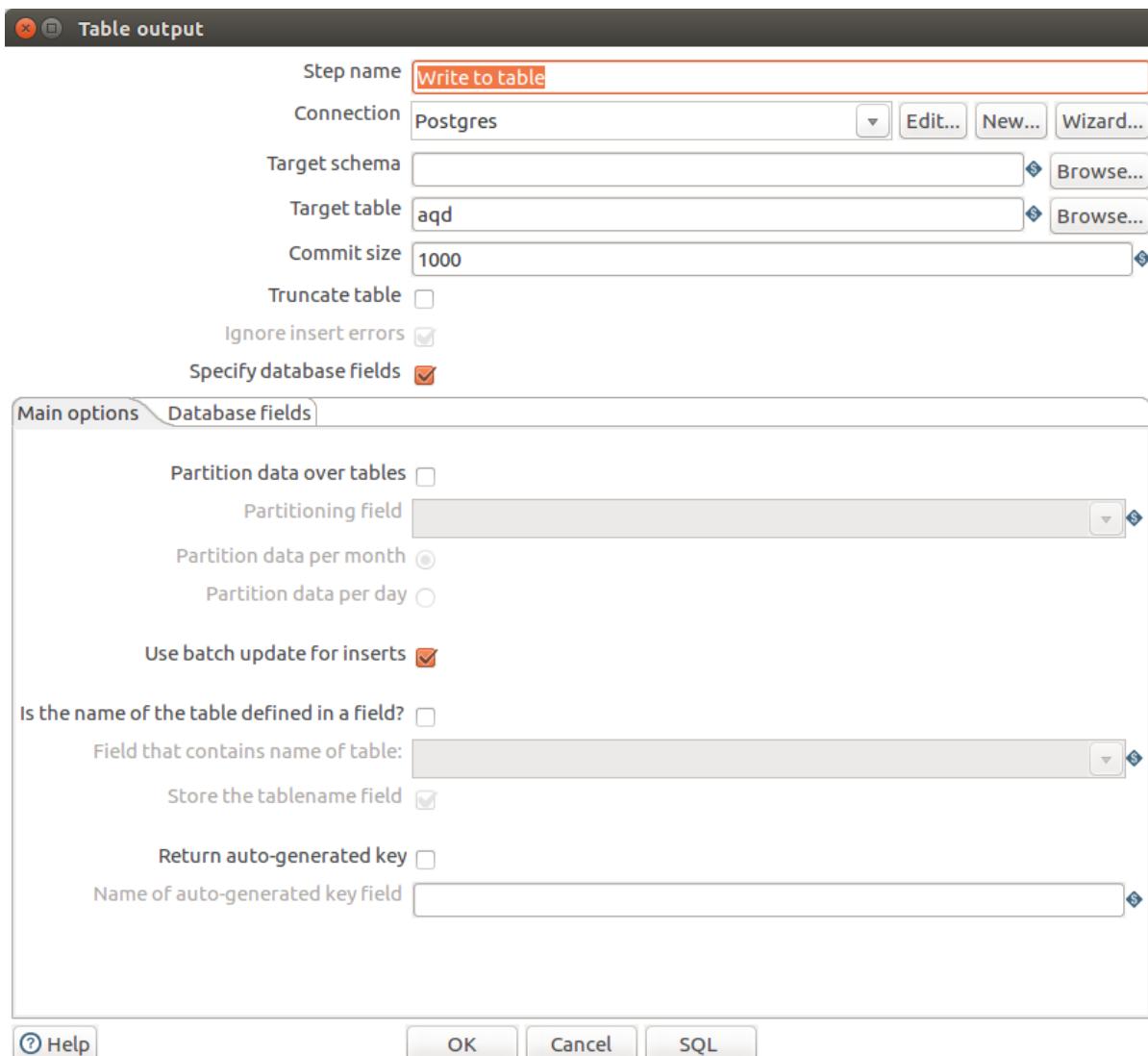
- d. Press OK to save and close the window

6. Drag the **Table Output** step to the canvas to write the selected values into a Postgres table
7. Create a hop between the **Select values** and **Table Output** steps and choose **Main output of step** from the popup
8. Double click on the **Table Output** step to specify its properties
 - a. Name: Write to table
 - b. Enter the following configuration.
Select the Postgres connection. If it does not exist in the drop-down list, you can create it yourself by using the **New...** button and entering the following configuration:



Password is postgres. Make sure you test the connection by selecting the **Test** button, so you are sure the connection is working.

After the connection, the rest can be specified:



As a last step, we can create the table in the database by selecting the SQL button.

c. Press OK to save and close the window

 The Table Output step allows you to load data into a database table. All database able to handle JDBC can be loaded over this step. Table Output is equivalent to the DML operator INSERT. This step provides configuration options for target table and a lot of housekeeping and/or performance-related options such as Commit Size and Use batch update for inserts.

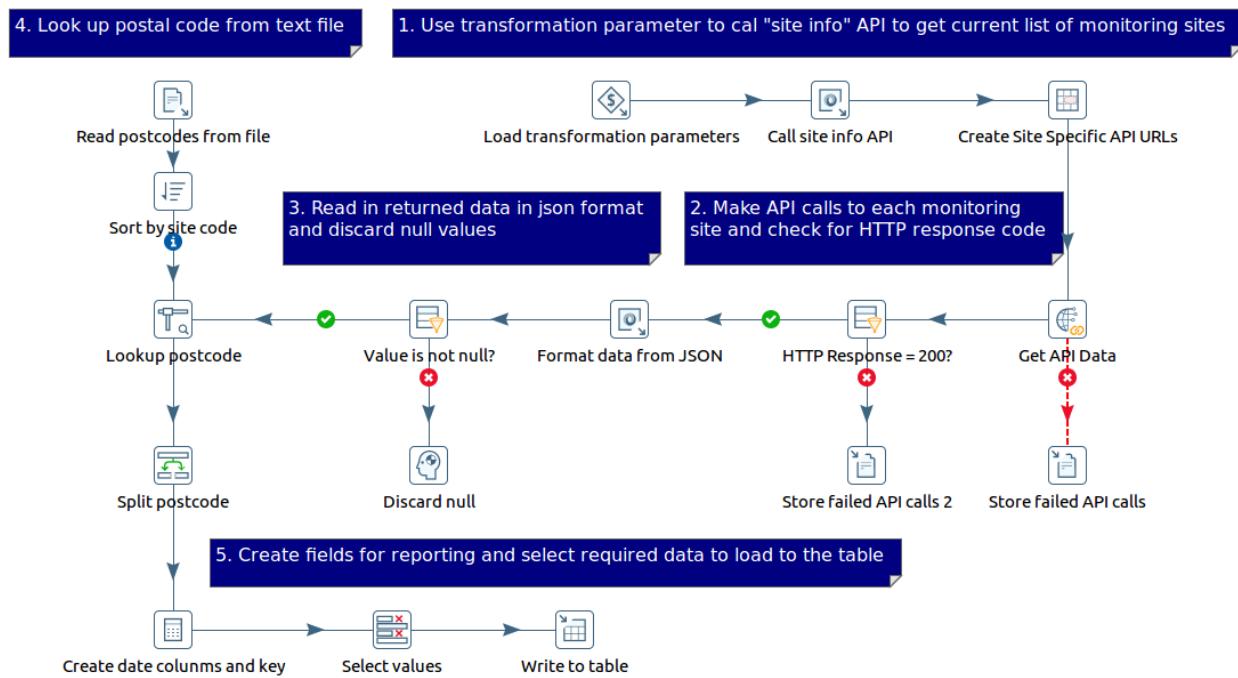
The SQL button let you create or alter the table in the database.

 The previous steps, Select values and Table Output, can be done in one step as well. Unless you are changing the metadata of the fields. The Table Output step has a tab Database fields where you can specify which fields should be loaded into the table.

This is a mapping between stream input fields and the database table columns. To make this possible the option Specify database fields should be enabled in the Table Output Step.

9. Save the transformation

And this should be your end-result of the transformation (without the comments; these are added for documentation):



Dimensional tables

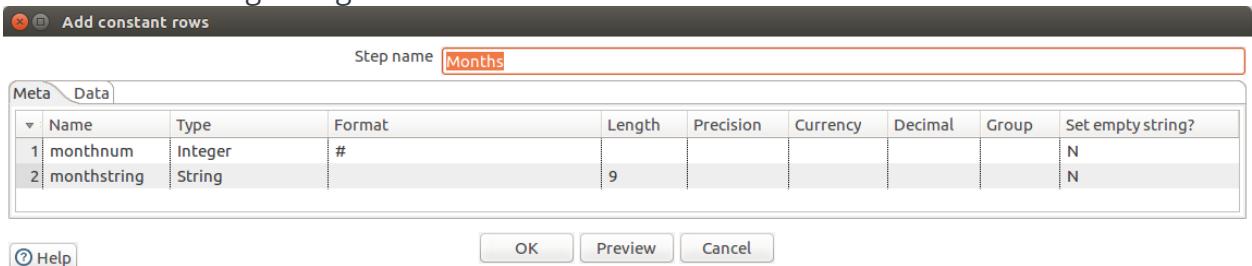
Our schema has its time dimension month and days in a numeric format. To show the names of the months and days we need additional dimension tables containing an ordinal column. We will create these tables via two very easy transformations.

 Another solution is to create a snowflake schema, in which the Month and Day table contain the ordinal column as well. See also the **Important aspects of the Data Source Wizard** chapter.

Months table

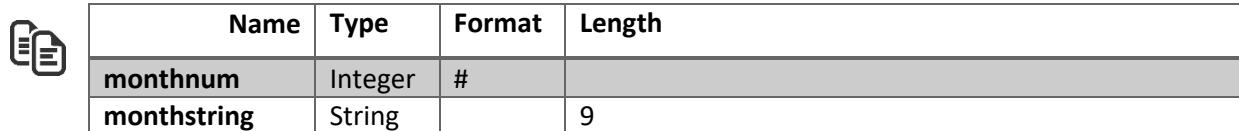
Let us start with the Month table. This table exists of 2 columns: a number and a name.

1. Create a new transformation
2. Save it as t_create_month_day_tables
3. Drag the **Data Grid** step to the canvas to specify the table layout and content
4. Double click on the **Data Grid** step to specify its properties
 - a. Name: Month
 - b. Enter the following configuration on the Meta tab:



The screenshot shows the 'Meta' tab of the Data Grid step configuration dialog. The 'Step name' is set to 'Months'. The table definition consists of two columns: 'monthnum' (Type: Integer, Format: #, Length: 9) and 'monthstring' (Type: String). The 'Data' tab is visible at the top right.

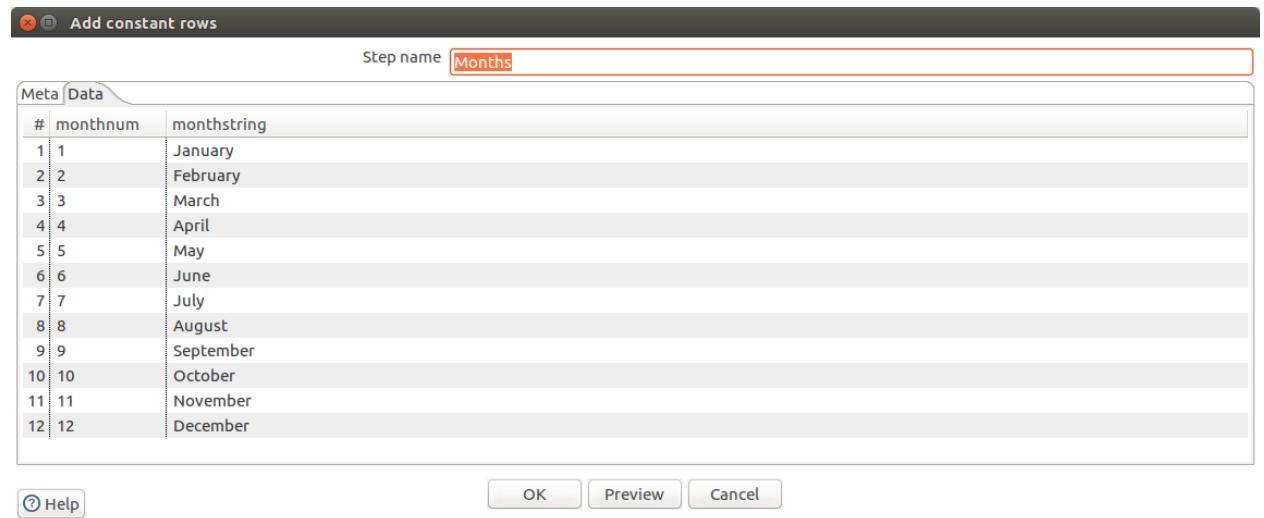
Name	Type	Format	Length	Precision	Currency	Decimal	Group	Set empty string?
1 monthnum	Integer	#	9					N
2 monthstring	String							N



The screenshot shows the 'Data' tab of the Data Grid step configuration dialog. It displays the same table structure as the Meta tab, with columns 'Name', 'Type', 'Format', and 'Length'.

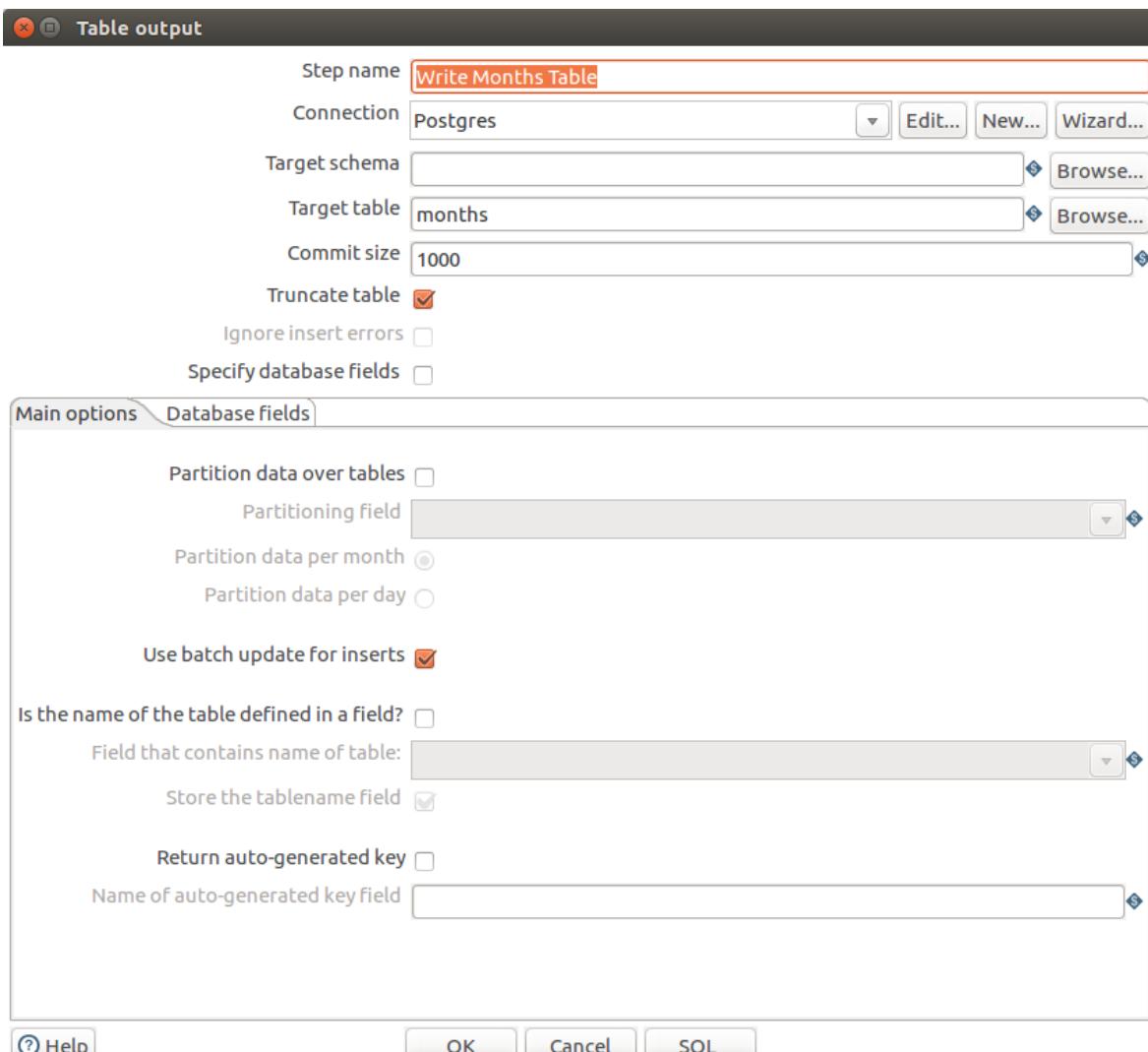
Name	Type	Format	Length
monthnum	Integer	#	
monthstring	String		9

- c. Enter the following configuration on the Data tab:



monthnum	monthstring	
1	January	
2	February	
3	March	
4	April	
5	May	
6	June	
7	July	
8	August	
9	September	
10	October	
11	November	
12	December	

- d. Press OK to save and close this window
- 5. Drag the **Table Output** step to the canvas to store the data in a database table
- 6. Create a hop between the **Data Grid** and **Table Output** steps
- 7. Double click on the **Table Output** step to specify its properties
 - a. Name: Write Months Table
 - b. Enter the following, leaving the specify database field option disabled, meaning that the table will be created with all fields from the transformation flow.



- c. Press the SQL button to create the table
- d. Press OK to save and close the window
- e. Save the transformation, which should look like this:



Days table

Creating the Days table is a same exercise.

1. Drag the **Data Grid** step to the canvas (*in the same transformation*) to specify the table layout and content
2. Double click on the **Data Grid** step to specify its properties
 - a. Name: Days

- b. Enter the following configuration on the Meta tab:

The screenshot shows the 'Add constant rows' step configuration window with the 'Meta' tab selected. The 'Step name' is set to 'Days'. The table contains two columns: 'Name' and 'Type'. Row 1 has 'weekdaynum' as Integer and '#' as Format. Row 2 has 'daystring' as String and '9' as Length. Buttons at the bottom include 'OK', 'Preview', and 'Cancel'.

Name	Type	Format	Length	Precision	Currency	Decimal	Group	Set empty string?
1 weekdaynum	Integer	#						N
2 daystring	String		9					N



Name	Type	Format	Length
weekdaynum	Integer	#	
daystring	String		9

- c. Enter the following configuration on the Data tab:

The screenshot shows the 'Add constant rows' step configuration window with the 'Data' tab selected. The 'Step name' is set to 'Days'. The table contains three columns: '#', 'weekdaynum', and 'daystring'. Rows 1 through 7 map weekday numbers to their corresponding days of the week. Buttons at the bottom include 'OK', 'Preview', and 'Cancel'.

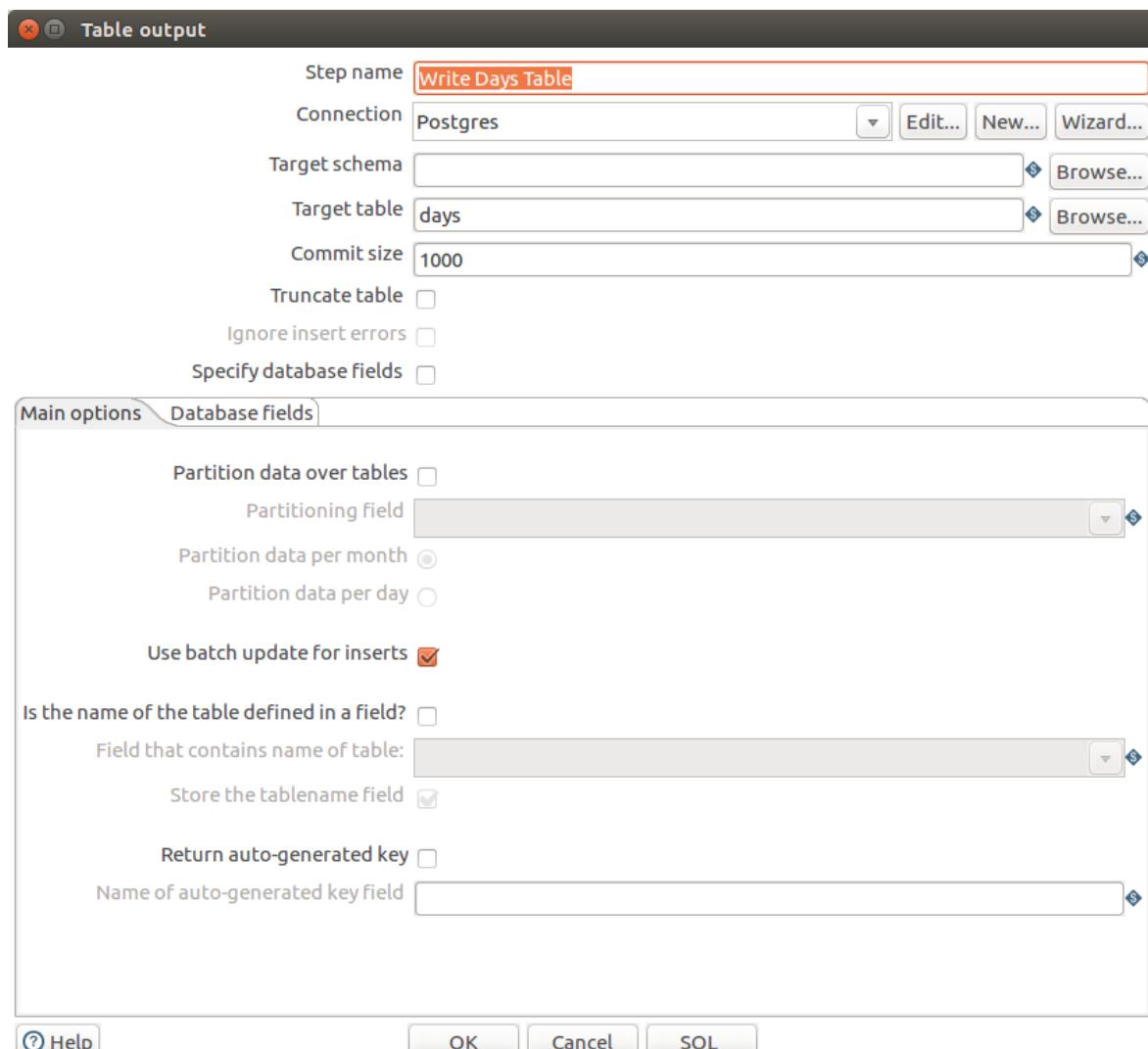
#	weekdaynum	daystring
1	1	Sunday
2	2	Monday
3	3	Tuesday
4	4	Wednesday
5	5	Thursday
6	6	Friday
7	7	Saturday



monthnum	monthstring
1	Sunday
2	Monday
3	Tuesday
4	Wednesday
5	Thursday
6	Friday
7	Saturday

- d. Press OK to save and close this window

3. Drag the **Table Output** step to the canvas to store the data in a database table
4. Create a hop between the **Data Grid** and **Table Output** steps
5. Double click on the **Table Output** step to specify its properties
 - a. Name: Write Days Table
 - b. Enter the following, leaving the specify database field option disabled, meaning that the table will be created with all fields from the transformation flow.



- c. Press the SQL button to create the table
- d. Press OK to save and close the window
- e. Save the transformation, which should look like this:



Creating Visualizations

Important aspects of the Data Source Wizard

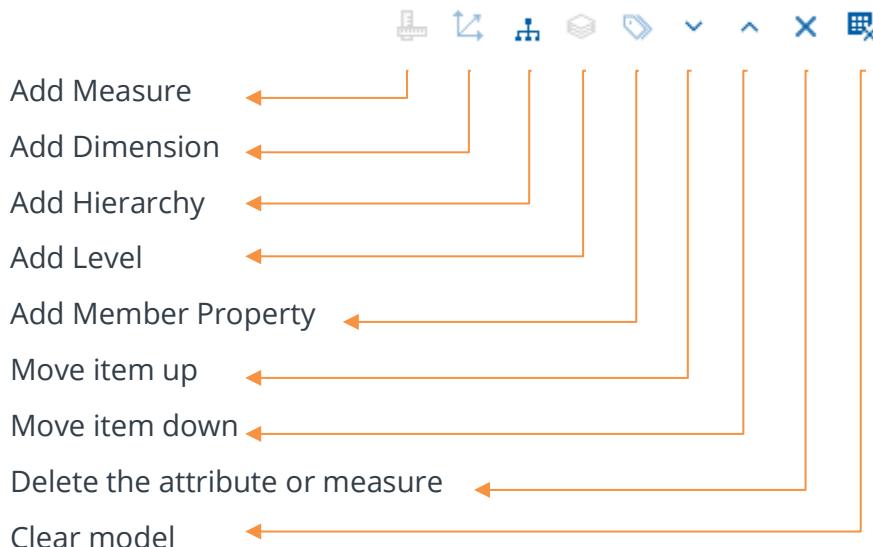
Before you can create Pentaho reports and analysis, the Pentaho Server needs to know where to find your data and how to connect to it, what data you want to use, how the data is structured, and what the data is like. The **Data Source Wizard** helps you define a data source that contains this information and guides you through the creation of your **relational or multi-dimensional data models** for use in creating reports and analysis.

This means that in the data source wizard a connection to a data source can be defined and a data model on that data source. The data models are semantic business models and are stored in the central **Pentaho Repository**. Such a semantic data model can be a part of the total physical data model and can be used by any number of reports or analysis.

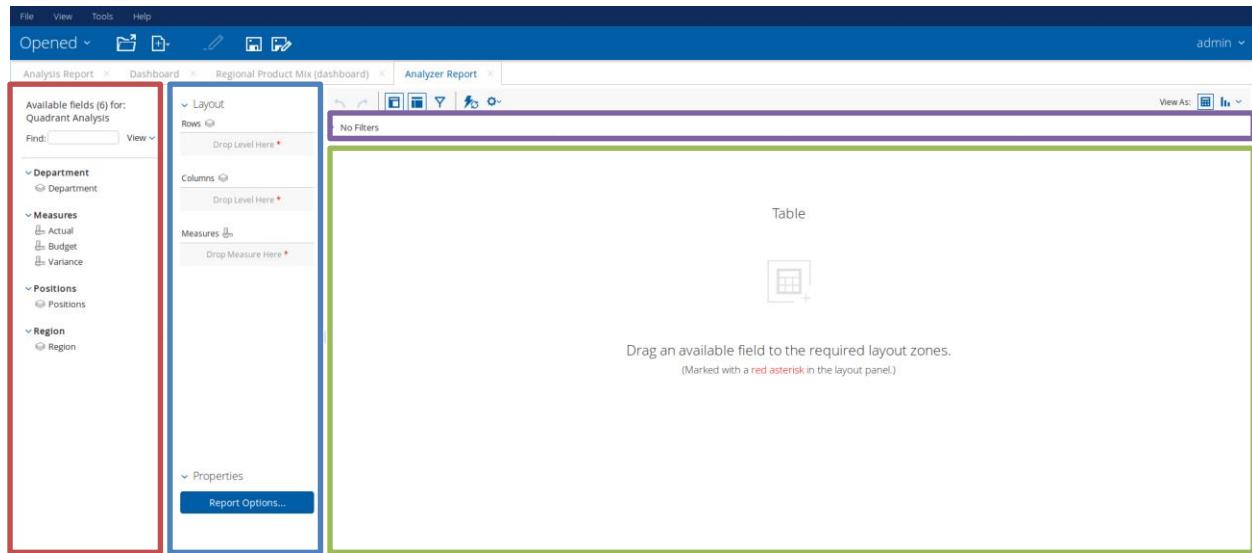
The Data Source Wizard only creates a **star schema** as a multi-dimensional model. **snowflake** schema's can be created with the **Pentaho Schema Workbench**. A best practice is to start with a star schema created in the Data Source Wizard, export it and import it in the Schema Workbench. After modifying it into a snowflake schema you can publish it to the Pentaho server again, where after it appears in the Data Source Wizard again.

After you define a data source, you can make it available to people who create reports and analysis so they can simply pick the data source they want to use from a list and begin creating their reports or analysis.

Icons used in the Data Source Model editor



Important aspects of Analyzer



Red Box: The Fields panel showing the semantic business schema for the analysis. This is a virtual dimensional schema on top of a relational database and is a subset of the relational schema. This semantic business schema can be used for one or more analyses. The Fields panel also shows the name of the semantic schema as Available Fields for <schema name>. For drilling down and up, the dimension in the schema should contain a hierarchy.

Blue Box: The Layout panel for the analysis. This will change depending on the content on the canvas. If the view is a table, then you will see rows, Columns, Measures and Report Options. If it is a graph/chart you will see X-Axis, Series, Measures, Multi-Chart and Properties, depending on the graph/chart type.

Green Box: The Report canvas. As soon as you start dragging attributes or metrics on the layout area, data will be fetched automatically and shown on the canvas.

 This automatically fetching can be switched off by clicking on the Disable auto refresh  icon. Now you can prepare your analysis and when it is finished you click again on the Disable auto refresh icon and the data will be fetched.

Although the schema contains metrics, you can define user-defined measures as well. To do this you use the metrics in the schema or already user-defined measures. You can define:

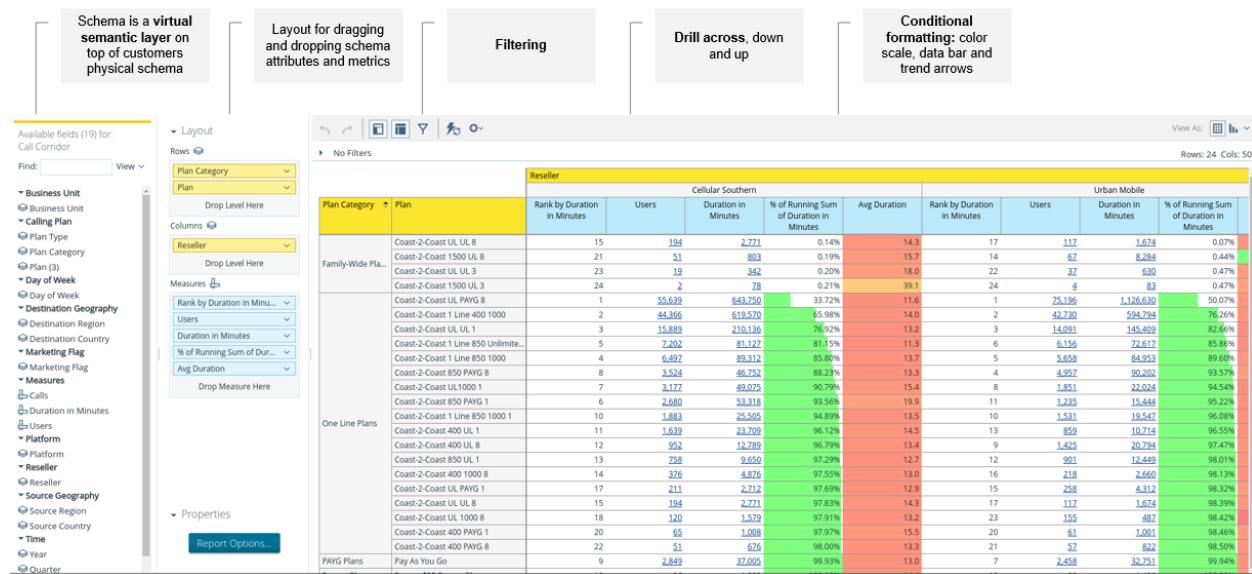
- % of
- Rank by
- Ranking Sum of
- % Running Sum of

- Trend, but you need a time dimension to create a trend measure
- Calculation

Purple Box: The Filters panel, where you can filter the content shown on the report canvas. Use the Filters panel to display a list of filters applied to the active report, or edit or delete filters.

In the right upper corner of the previous figure you can see the  icons to switch between the table and a chart. We have about 15 out-of-the box charts, but since we are 100% Java and open you can add additional charts. These can be downloaded from the Pentaho Website or from the Pentaho Marketplace (D3js). These are the easy ones and work immediately. You can also download charts from <https://www.highcharts.com/> or <http://www.fusioncharts.com/>, but after the download you need to do a minor tweaking of the charts, so it can talk to our classes and then save it in a Pentaho folder.

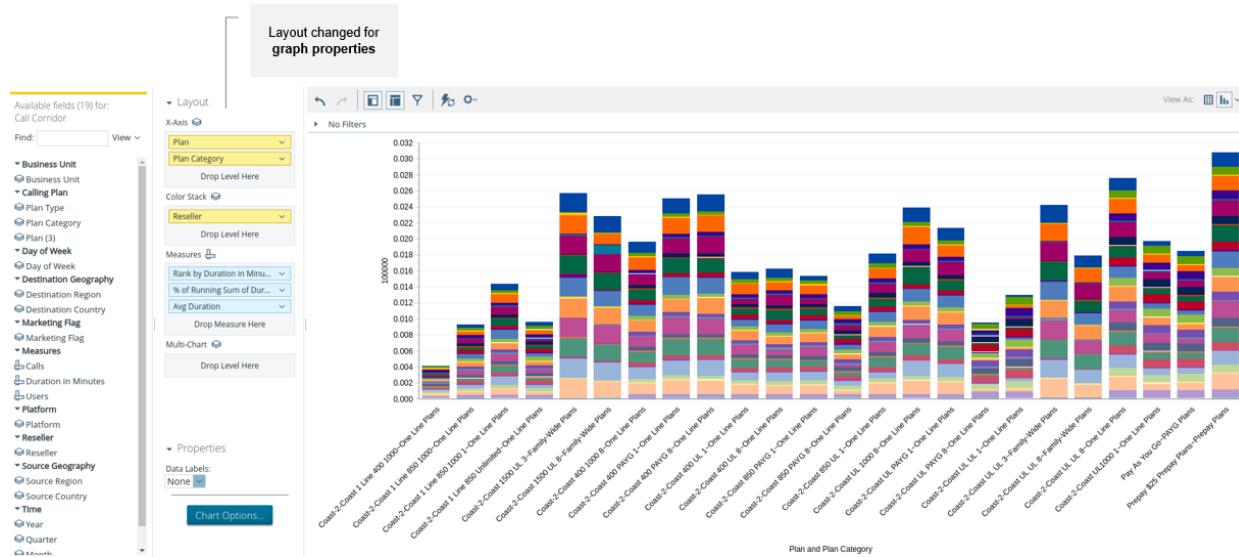
Example of an analysis with remarks (not part of the analysis):



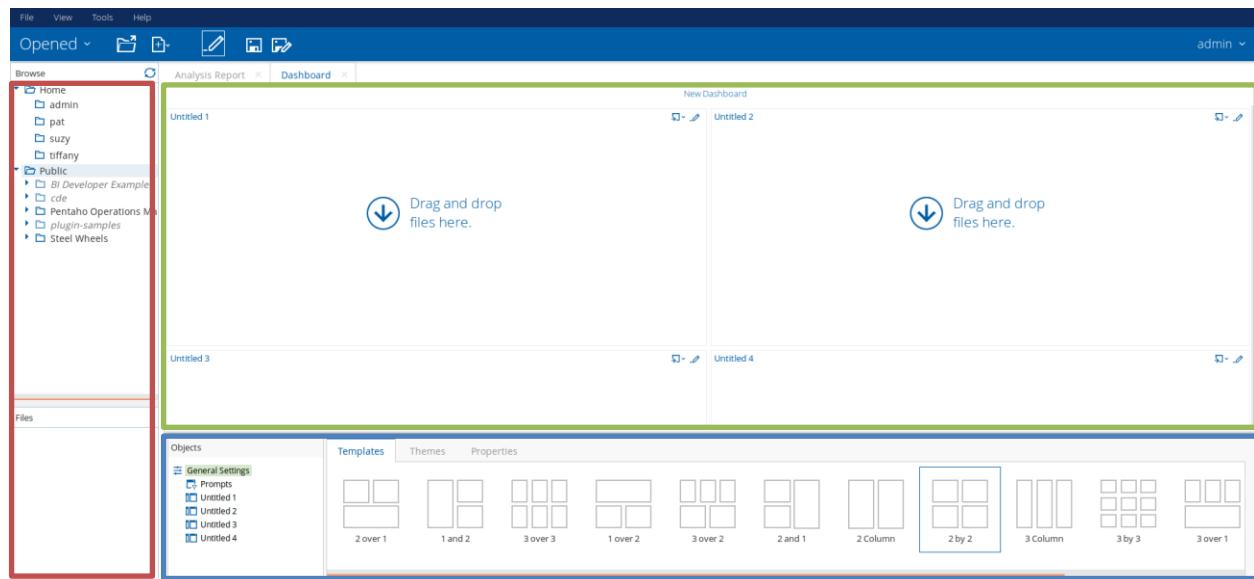
The screenshot illustrates the Pentaho LAQ interface with several key features highlighted:

- Schema is a virtual semantic layer on top of customer physical schema:** A callout points to the left sidebar where the schema browser is visible.
- Layout for dragging and dropping schema attributes and metrics:** A callout points to the layout pane where users can drop fields into rows and columns.
- Filtering:** A callout points to the filtering pane at the top of the report canvas.
- Drill across, down and up:** A callout points to the drill-down feature in the report canvas.
- Conditional formatting: color scale, data bar and trend arrows:** A callout points to the data table, which uses conditional formatting to highlight specific values.

The main report canvas displays a data table titled "Reseller" with two sections: "Cellular Southern" and "Urban Mobile". The table includes columns for Plan Category, Plan, Rank by Duration in Minutes, Users, Duration in Minutes, % of Running Sum of Duration in Minutes, Avg Duration, and similar metrics for the Urban Mobile section. The data shows various plan categories and their performance metrics across different regions and user counts.



Important aspects of Dashboard Designer



Red Box (Browse and Files Panel): The usable components (reports) in the central Pentaho Repository.

Pentaho has a central Repository that stores the objects created in the Pentaho Platform. These objects are:

- Business Analytics objects: Interactive Reports, Analyses, Dashboards, schedules and connectivity objects
- Data Integration objects: Jobs, Transformations, schedules, versioning, cluster information and connectivity objects

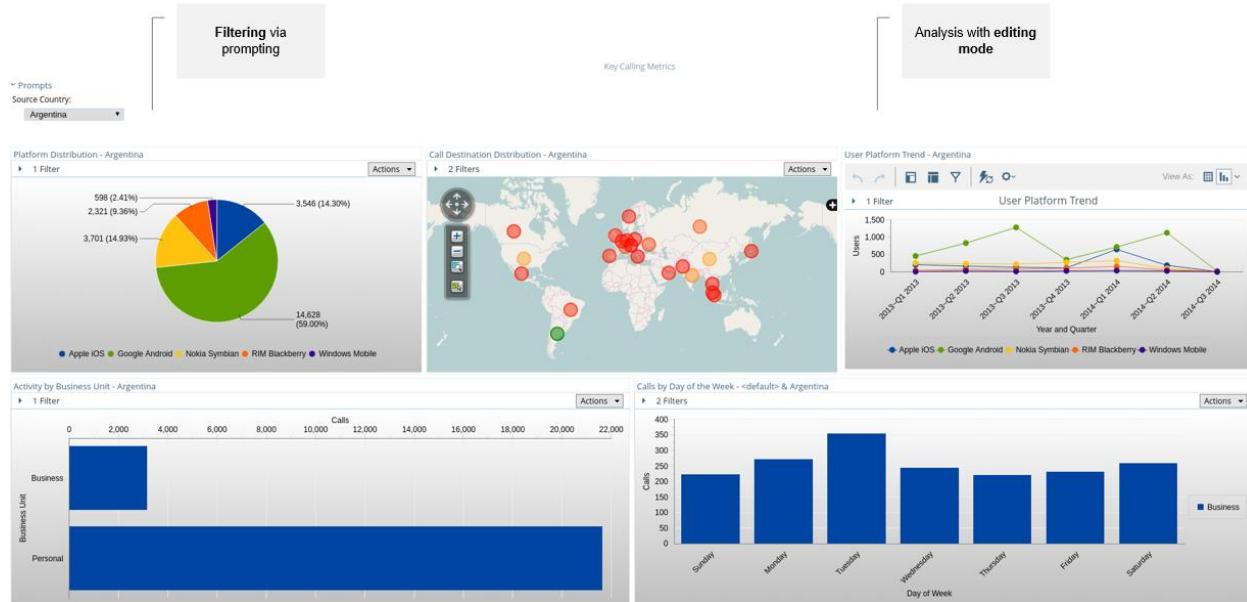
Locate your files using the **Browse** and **Files** panels, and add them to dashboards.

Green Box (Dashboard Canvas): The canvas where you can drag and drop the usable components in the central Repository on the panes. Showing a dynamic view of your Dashboard as you work to build it. The look of your dashboard refreshes as you add content from the Browse Folders and Files panels, and work with the Prompts or Objects panels.

Blue Box (Objects Panel): The Objects, Templates, Themes used in the dashboard and Properties of the dashboard.

- **Objects:** General settings (being the templates, themes and properties), Prompts and content of the dashboard panes.
 - **Templates:** there are 13 out of the box templates, going from 1 pane to 9 panes you can use. You also can add your own, since Pentaho is 100% Java and embeddable. You can write your own template XML file and save it in the correct Pentaho folder, being /pentaho/server/pentaho-server/pentaho-solutions/system/dashboards/templates/xul. Keep in mind that for more than 9 panes in the dashboard you need a big screen to make it readable.
 - **Themes:** there are 8 out of the box themes you can use. Like with templates, you also can add your own. You can write your own theme file and save it in the correct Pentaho folder, /pentaho/server/pentaho-server/pentaho-solutions/system/dashboards/themes. These can be themes per end-customers, that will be used by each individual customer via multi-tenancy. In this way, they can embed it in their own portal as if it is part of that portal (white labeling).
 - **Properties:** Title of the dashboard and the refresh interval in seconds for scheduling. Instead of scheduling the complete dashboard, you can also schedule individual objects (in the panes). The properties pane also includes a **Resize** button to resize the panes.
 - **Prompts:** to define prompts for the Prompt bar in the top of the dashboard. Prompt types available are drop-down, list, radio buttons, checkbox, buttons, text field and a date picker (incl. from to dates). A prompt is in fact a filter to one or more panes. For prompts to work the filter in the analysis should have defined a parameter.
 - **Content of the dashboard panes:** the predefined analytics in the dashboard panes. When creating a new dashboard, the panes have no objects and you can drag a report on the pane to give it content. Then click on the Untitled # object to give it a proper name. You can choose to name it according the content name shown here. This name will be shown as a title on top of the pane. You also can specify a refresh interval in seconds here (for scheduling). You can define the parameters here that will be used in combination with the prompts. Linking content between panes is possible by enabling a linking field. For linking to work the filter in the analysis linked to should have defined a parameter.

Example of a dashboard with remarks (not part of the dashboard):



Get started

Now we have our data we can start our analysis.

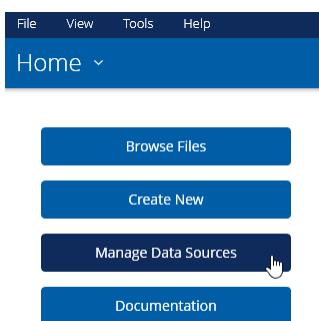
- From a web browser, connect to the Pentaho User Console at <http://localhost:8080> using the following credentials:
 - Username: admin
 - Password: password



We are now using a web based thin client. Please be careful not to refresh your page without saving your work, otherwise you could lose your progress.

Creating the data source

- Click on Home > Manage Data Sources



- Press **New Data Source** in the **Manage Data Sources** window

The screenshot shows a 'Manage Data Sources' interface. At the top right is a blue 'New Data Source' button with a cursor pointing at it. Below is a table with columns 'Datasource' and 'Type'. The data includes:

Datasource	Type
AgileBI	JDBC
Audit	JDBC
ba-pom	Metadata
live_logging_info	JDBC
PDI_Operations_Mart	Metadata
pentaho_operations_mart	Analysis

At the bottom right is a blue 'Close' button.

4. Give the to be created **semantic business model** the name Air Quality Data and select Database Table(s) as **Source Type**.

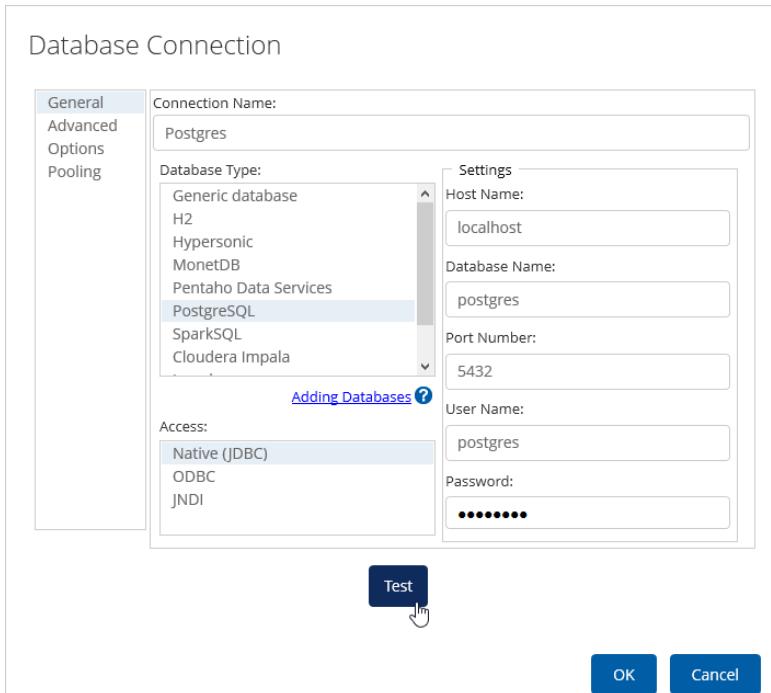
 The user creating the model need to have read permissions on the metadata of the database else no tables will be shown when using the Database Table(s) as Source Type.

Click on the plus sign to add a new **database connection**

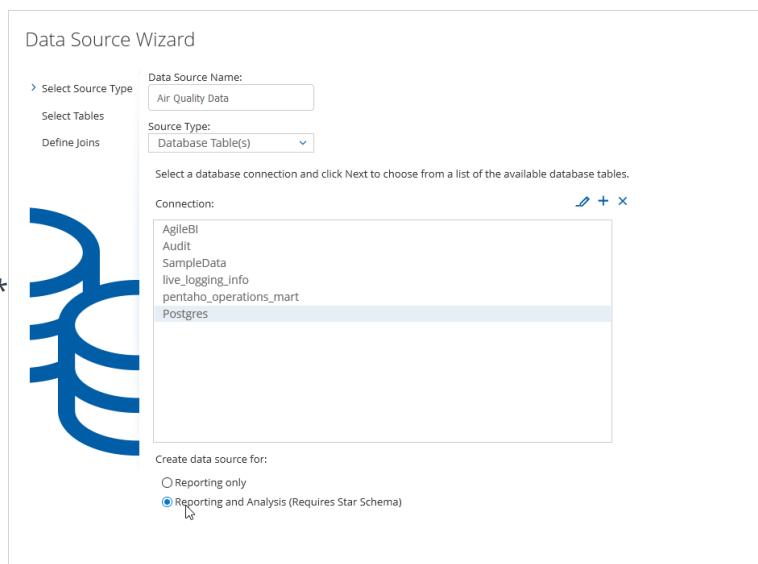
The screenshot shows the 'Data Source Wizard' with the 'Select Source Type' step selected. On the left is a decorative blue wavy graphic. The main area has fields for 'Data Source Name' (set to 'Air Quality Data') and 'Source Type' (set to 'Database Table(s)'). Below is a note: 'Select a database connection and click Next to choose from a list of the available database tables.' A 'Connection:' dropdown lists the available databases: AgileBI, Audit, SampleData, live_logging_info, and pentaho_operations_mart. To the right of the list is an 'Add Connection' button. At the bottom, there's a note: 'Create data source for:' with radio buttons for 'Reporting only' (selected) and 'Reporting and Analysis (Requires Star Schema)'.

5. Add the following properties in the **Database Connection** window:

- a. Connection Name: Postgres
- b. Add the following properties. Password for the user postgres is postgres.

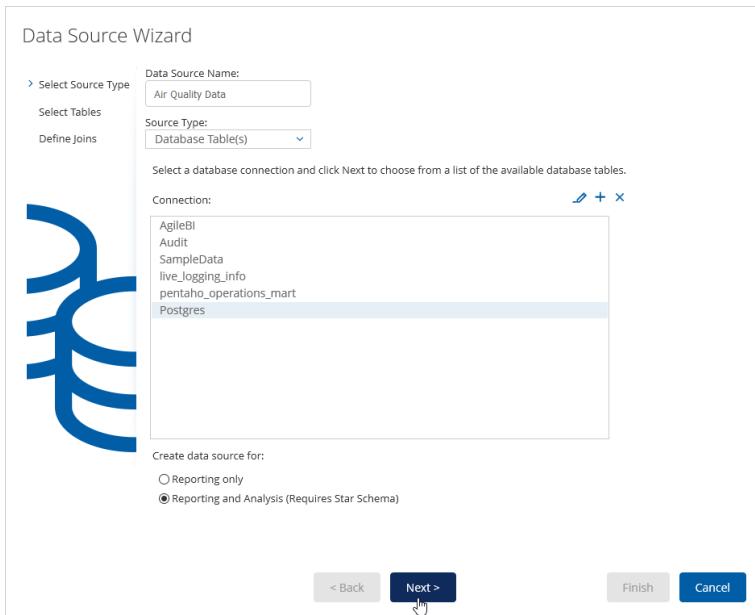


6. Press **Test** to check if you get a connection
7. Press **OK** to save and close the **Database Connection** window
8. Once we have the database connection, we select the just created connection (Postgres) and click on the option Reporting and Analysis (Requires Star Schema).

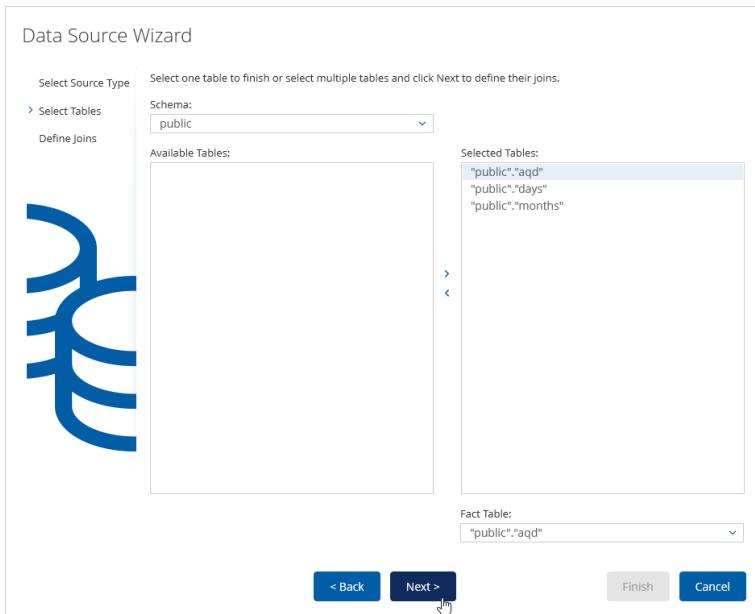


 The option **Reporting only** creates a relational model for interactive reporting. The **Reporting and Analysis** option creates a relation model as well as a multi-dimensional model for analysis. The models created are semantic business models.

9. Press **Next**



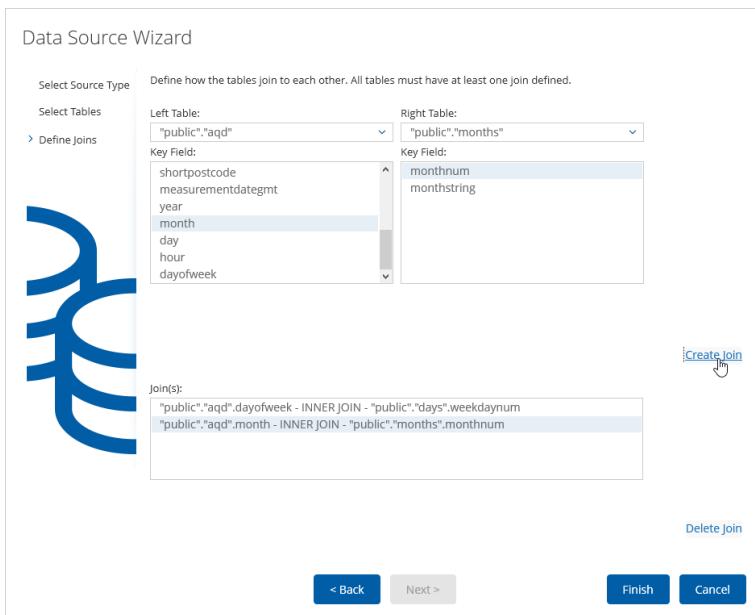
10. Select all tables and move them to the **Selected Tables** area and press **Next** when done



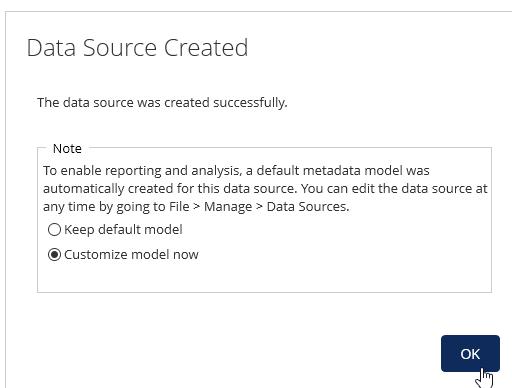
11. Create the joins between:

- Chose aqd.dayofweek and days.weekdaynum and press the **Create join** link
- Chose aqd.month and months.monthnum and press the **Create join** link

Do not touch the left table, since this is the fact table. Just change the right table for creating the joins.



12. Press **Finish** when finished. The Finish button is greyed out until the necessary joins are created (number of tables minus one)
13. A window pops up asking to keep or customize the model. Normally you want to customize it. Especially a multi-dimensional model, since you need to indicate the time and geo dimensions.
Hence select the **Customize model now** option.



In the analysis tab you will see the generated multi-dimensional semantic schema. This needs to be checked on correctness. If you expand the measures you will see also numeric attributes (the generator cannot make a difference between attributes and metrics, hence it considers all numeric fields as measures).

Data Source Model Editor

Available

Analysis Reporting

Measures

- Day
- Dayofweek
- Hour
- Lat
- Lon
- Month
- Value
- Year

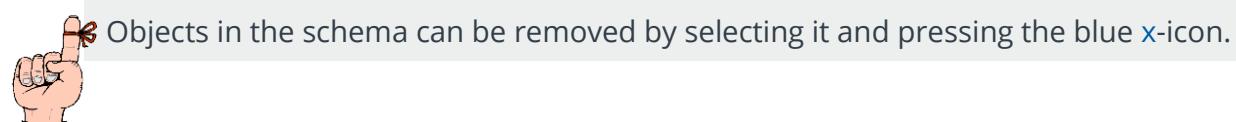
Dimensions

- Days
- Days
- Months

Properties

OK Cancel

14. Remove all **measures** except the Value one. DO NOT touch the Dimensions yet.



15. Select the measure **Value** and select AVERAGE as **Default Aggregation** and **#.#** as the **Format**

Data Source Model Editor

Available

Analysis Reporting

Measures

- Value

Dimensions

Properties

Source Column: Value

Details

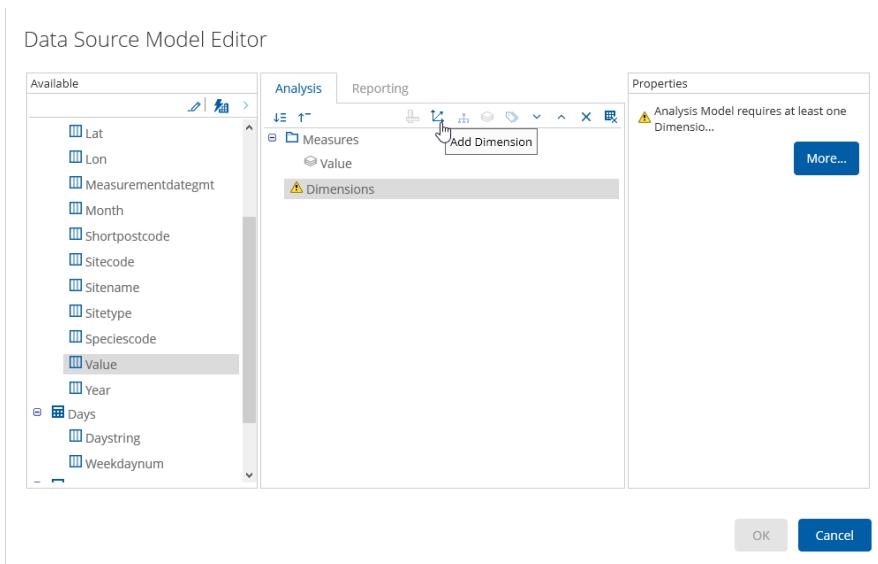
Display Name: Value

Default Aggregation: AVERAGE

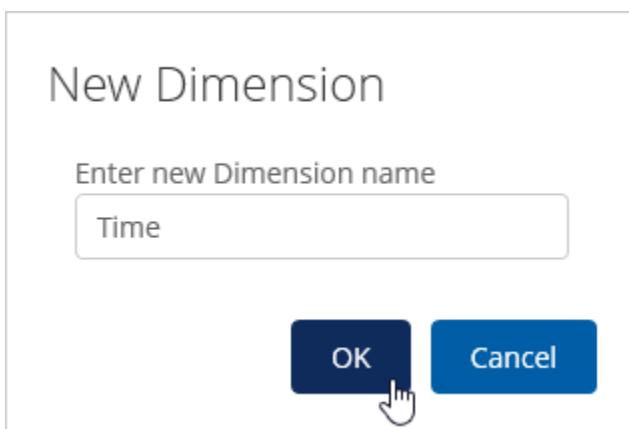
Format: #.#

OK Cancel

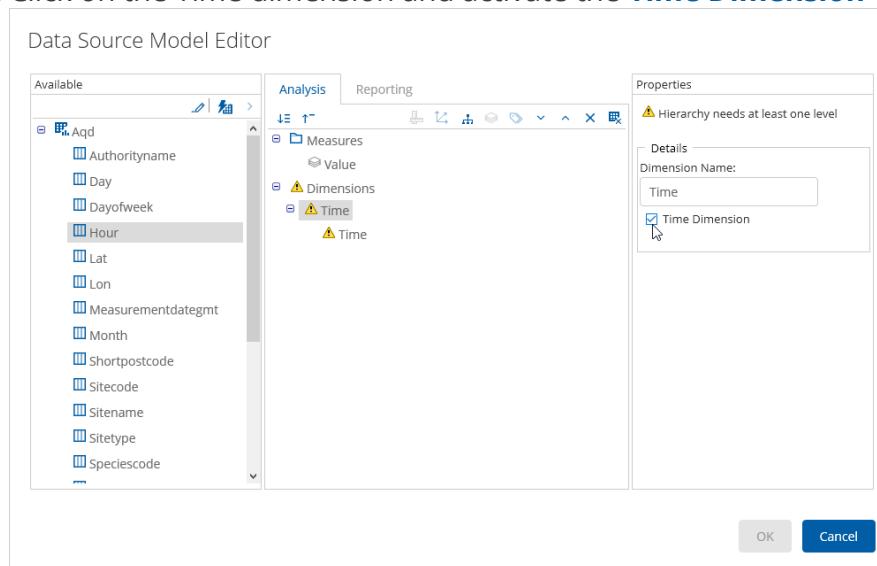
16. Click on the Dimensions in the Analysis tab and create a new dimension by clicking on **Add Dimension** icon (see screenshot)



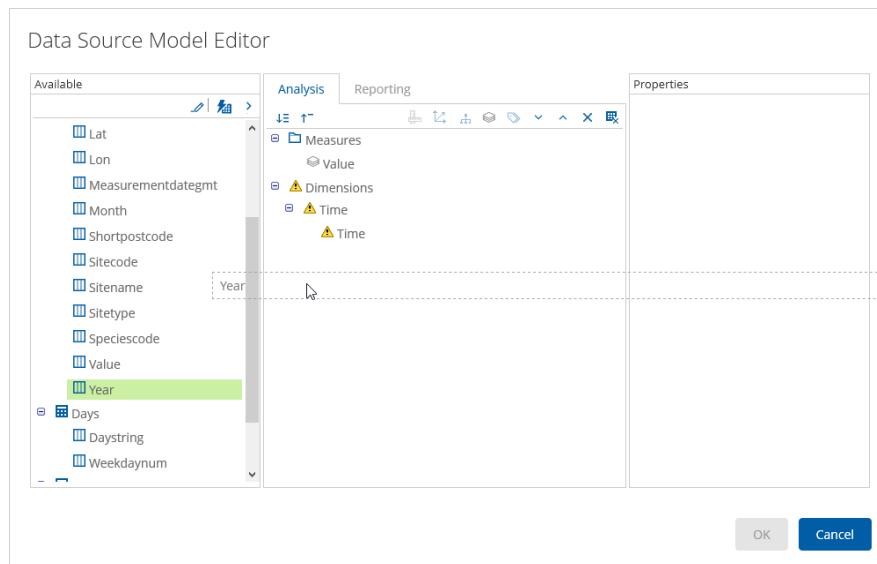
17. Enter Time as the name for the new dimension and press **OK**

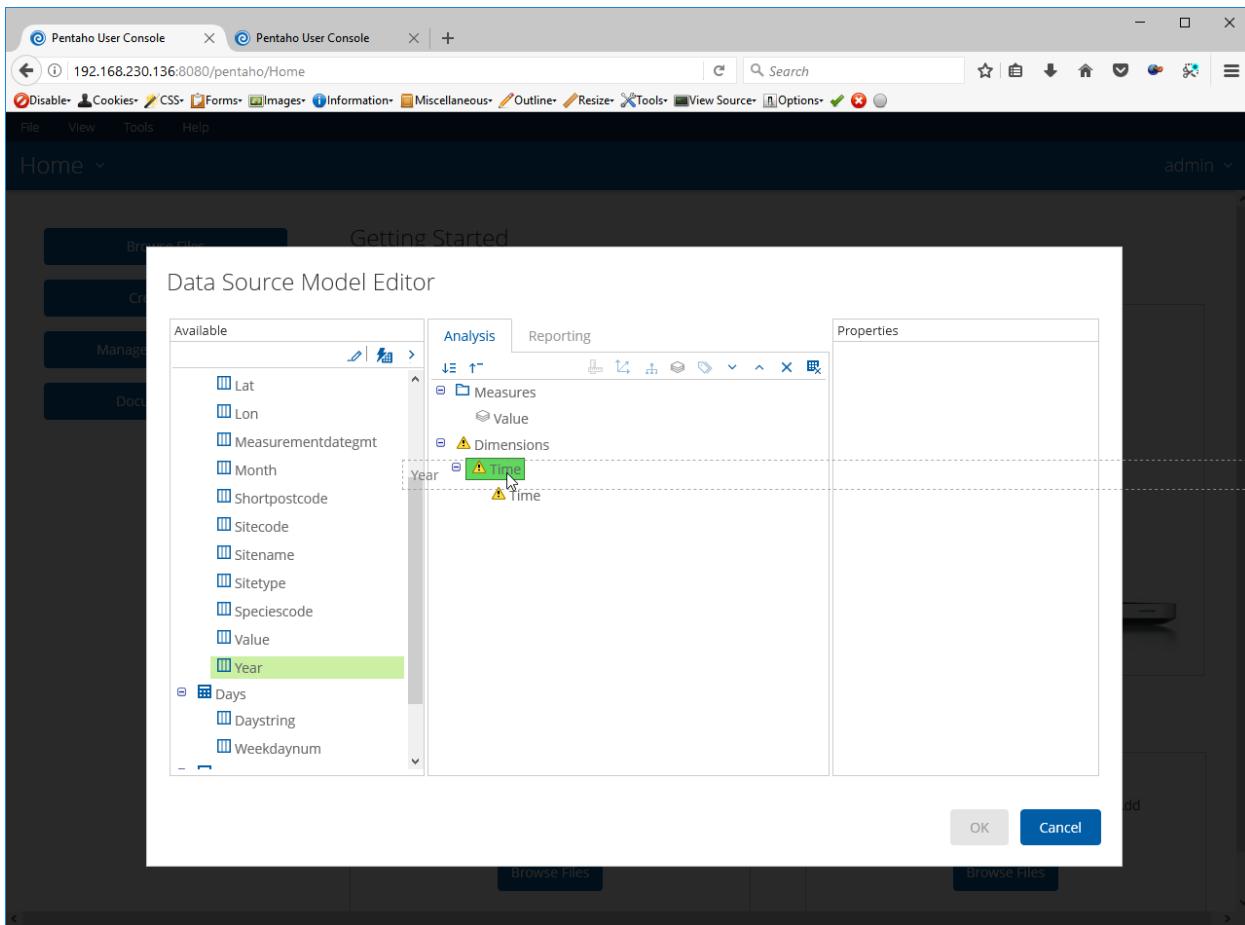


18. Click on the Time dimension and activate the **Time Dimension** option in the Properties



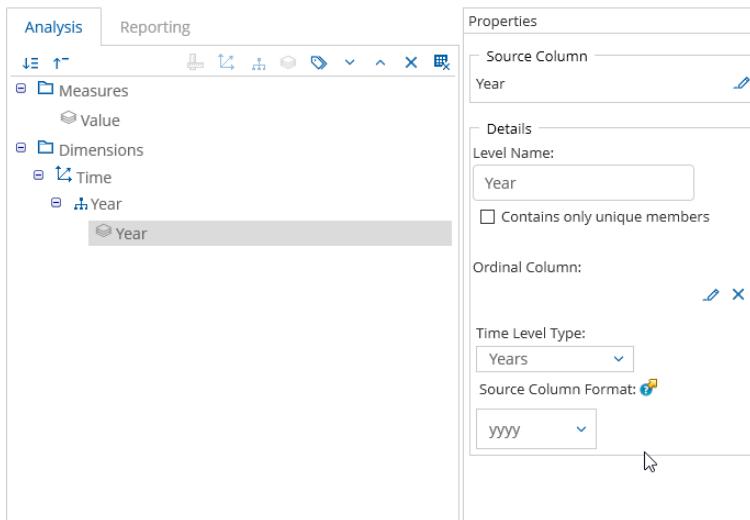
19. Now drag the attribute year on top of the Time **dimension** (as seen in the full screen screenshot below). The Time **dimension** should change into green.





20. Delete the **Time level** (be careful not to delete the dimension!)

21. For the Year level, specify the proper **Time Level** and **Format** properties:



22. Add Month, Day and Hour to the Time **dimension**, making sure the hierarchy is Year, Month, Day and Hour. Specify the correct **Time Level** and **Format**, being:

- Month: Months, MM
- Day: Days, dd
- Hour: Hours, HH

The screenshot shows the Pentaho Analysis interface. On the left, the Dimensions tree is expanded, showing Measures, Dimensions, Time, Year, Month, Day, and Hour. The 'Hour' node under 'Time' is selected. On the right, the Properties panel is open for the 'Hour' dimension. It shows the Source Column as 'Hour', the Level Name as 'Hour', and the Time Level Type as 'Hours'. The Source Column Format is set to 'H'.

 In an analysis, the hierarchy can be used for drilling up and down levels.

The Days and Months dimension should already be in the model.

23. Rename the **Months** Dimension into **Month Name** in the **Properties**

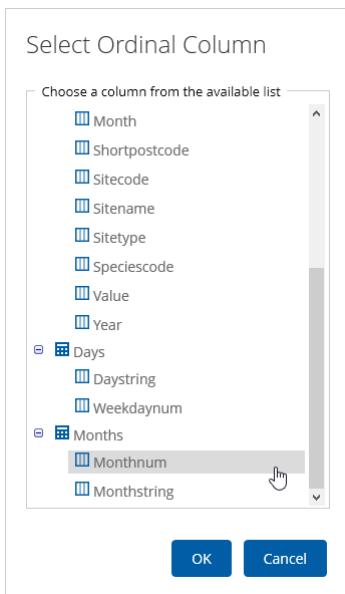
24. Check the **Time Dimension** flag

The screenshot shows the Pentaho Analysis interface. The Dimensions tree now shows 'Month Name' instead of 'Months'. The Properties panel for 'Month Name' shows the Dimension Name as 'Month Name' and the Time Dimension checkbox is checked.

25. Do the same for the Monthstring **hierarchy** and **level**

26. Click on the Monthname **level** and click on the pencil icon next to Ordinal Column. Choose Months.Monthnum as the **Source Column**

 Ordinal columns are used to alter the natural order of a level when fetching members from a database. The ordinal column is always a column with numeric values, used for presenting the non-ordinal column containing text values in the correct order.



27. Choose Months as **Time Level** and MMM for **Format**

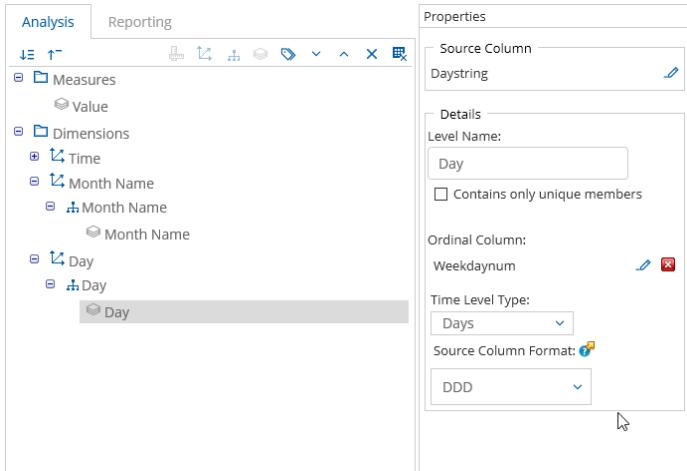
The screenshot shows the Pentaho Analysis interface with the 'Analysis' tab selected. On the left, the tree view shows 'Measures' and 'Dimensions'. Under 'Dimensions', there are 'Time', 'Month Name', and another 'Month Name' node. The second 'Month Name' node is selected and highlighted with a gray background. On the right, the 'Properties' panel is open for this selection. The properties are as follows:

- Source Column:** Monthstring
- Details** (Level Name: Month Name, Contains only unique members checked)
- Ordinal Column:** Monthnum
- Time Level Type:** Months
- Source Column Format:** MMM

28. Rename the Days **dimension** into Day in the **properties**

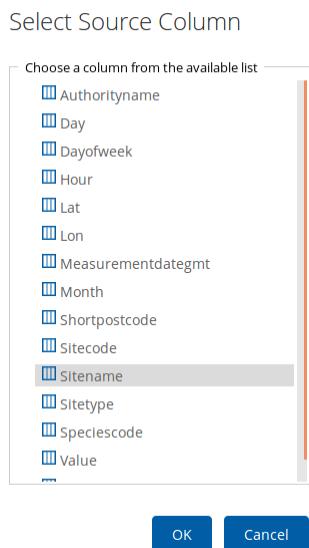
29. Do the same with the Weekdaynum **hierarchy** and **level**. Rename them into Day.

30. Click on the Day **level** and click on the pencil icon next to Ordinal Column. Choose Days.Weekdaynum as the **Source Column**

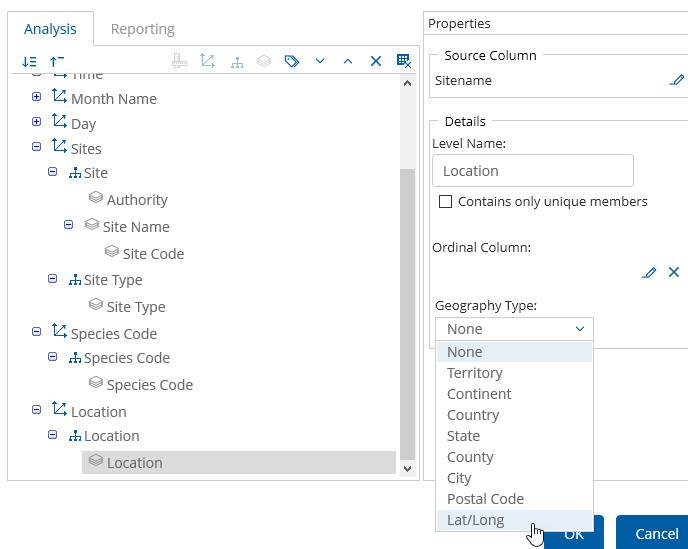


31. Click on the Dimensions in the Analysis tab and create a new dimension by clicking on **Add Dimension** icon
32. Call the new dimension Sites
33. Drag Authority, Sitename and Sitetype on the sites **dimension** to create a **hierarchy**.
34. Drag Sitecode on Sitename to create a second **level**
35. Remove the Sites **hierarchy** with the yellow exclamation sign
36. Drag the Species Code onto the Dimensions folder to create a new hierarchy and label it appropriately
 - Species Code**
 - Species Code**
 - Species Code**
37. Click on the Dimensions in the Analysis tab and create a new dimension by clicking on **Add Dimension** icon
38. Call the new dimension Location
39. Add a **level** named Location to the Location **hierarchy**
40. Click on the Location **level** and notice the yellow exclamation signs. To correct this, click on the latitude **level** and press the **Fix Missing column** button.

41. Select the Sitename attribute from the **Select Source Column** window and press **OK**

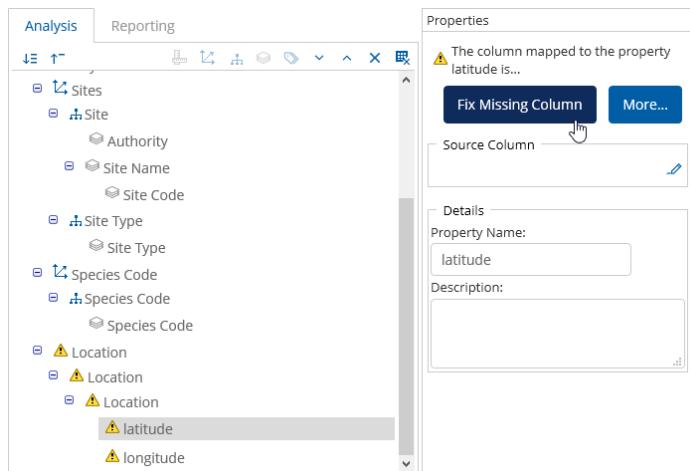


42. Click again on the Location **level** and chose the correct **Geography Type**, being Lat/Long. This is needed to plot the Location on a geomap.

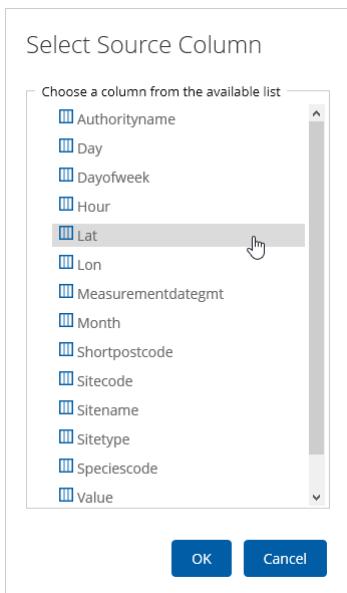


 The Latitude/Longitude level is transparent to the end-user. It need to be filled for plotting the level it belongs to on the geomap. This means that the Sitename level can have the Latitude/Longitude level as well (see screenshot above).

43. Expand the Location **level** and notice the yellow exclamation signs. To correct this, click on the latitude **level** and press the **Fix Missing column** button.



44. Select the Lat attribute.



45. Press **OK** to save and close the **Select Source Column** window

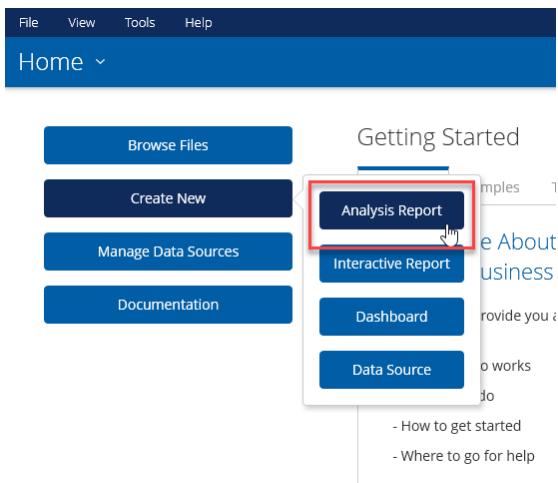
46. Do the same for the longitude **level** and chose Lon as source column

47. Press **Ok** to save and close the **Data Source Model Editor**

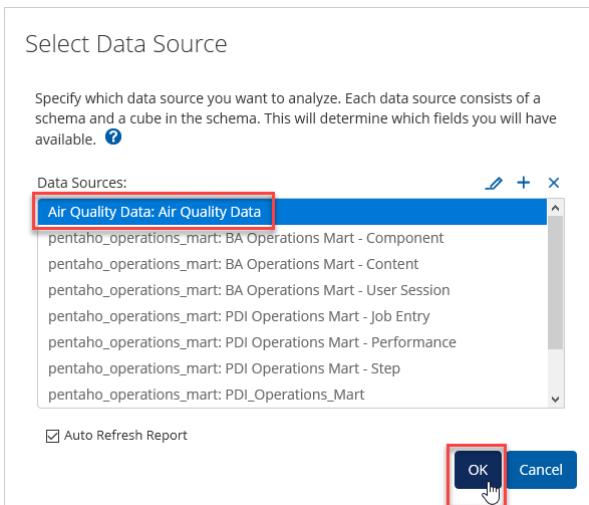
Creating the Average Pollutant by Hour Analysis

First, we are going to create the Average Pollutant by Hour analysis to view the average readings for each species code by hour of the day.

1. In the Pentaho User Console, click on the **Create New > Analysis Report** buttons



2. Choose in the **Select Data Source** popup window the Air Quality Data data source.



The Analyzer Editor should popup. In the left side frame, the semantic data model is depicted. The attributes and metrics can be viewed in several ways if clicked on the **View** next to the **Find** field.

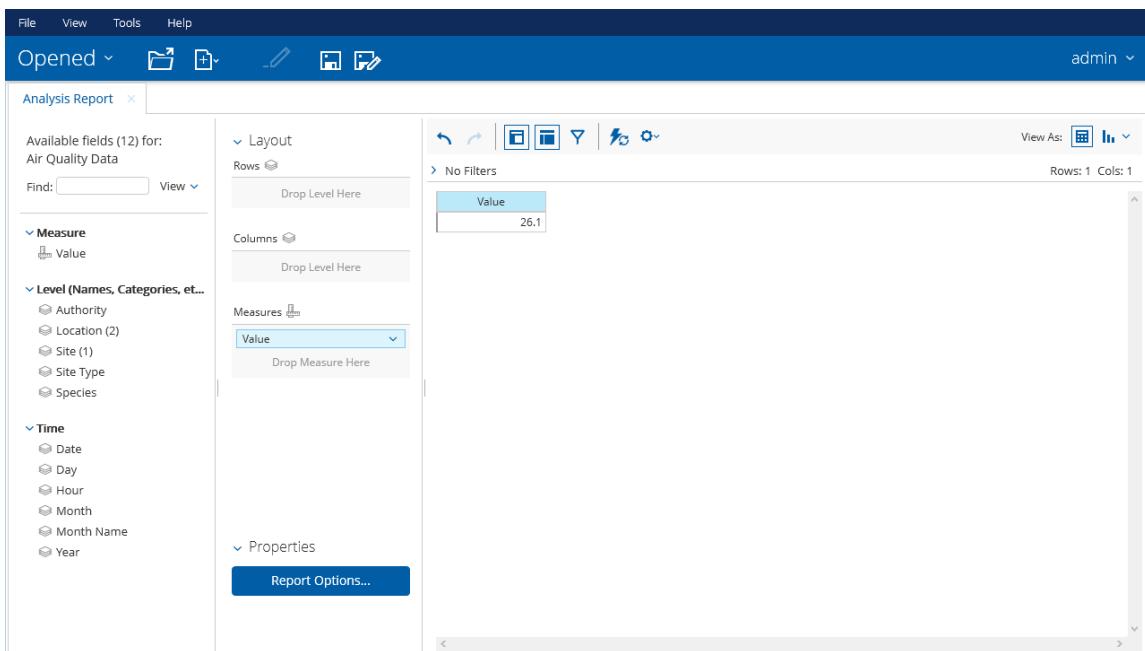
Select the option **Measure - Level - Time** or chose a view you like.

The screenshot shows the Pentaho Analyzer interface with the 'Analysis Report' tab selected. The left sidebar lists available fields for 'Air Quality Data' categorized by Authority, Date, Day, Hour, Location, Measures, Month, and Month Name. The 'Layout' panel on the right allows dragging fields into Rows and Columns. A red box highlights the 'View' dropdown menu. Another red box highlights the 'Measure - Level - Time' item in the 'by Category' list under the 'Layout' section.

3. Drag the **Value** metric to the Measures in the Layout area

The screenshot shows the Pentaho Analyzer interface with the 'Analysis Report' tab selected. The left sidebar lists available fields for 'Air Quality Data' categorized by Measure, Level, Time, and others. The 'Layout' panel on the right allows dragging fields into Rows and Columns. A red arrow points from the 'Value' metric in the 'Available fields' list to the 'Drop Measure Here' area in the 'Measures' section of the 'Layout' panel.

Notice that Analyzer is going to fetch the data immediately (see screenshot below).



The screenshot shows the Pentaho Analysis Report interface. On the left, there's a sidebar with a tree view of available fields: Measure (Value), Level (Names, Categories, etc...), and Time (Date, Day, Hour, Month, Month Name, Year). The main area is divided into three sections: Layout, Rows, and Measures. In the Rows section, there's a 'Value' field with the value '26.1'. In the Measures section, there's also a 'Value' field. At the bottom right of the main area, there's a 'Report Options...' button.



Instead of dragging, you can also double click the attribute or metric to be placed automatically on the proper area. This is for attribute the Rows area and for metrics the Measures area. Double clicking the attributes will preserve the hierarchy in the Layout. Manually you must do this yourself. You always can rearrange the attributes over the different subareas.



Sometimes, a field may show up as a dimension and as a measure in a data model. Be sure you pull the correct one into your report.

4. Drag the **Hour** attribute on to the X-Axis and the **Species Code** attribute to the Series in the Layout area.

Available fields (12) for: Air Quality Data

Find: View

Measure Value

Level (Names, Categories, etc.)

- Authority
- Location (2)
- Site (1)
- Site Type
- Species

Time

- Date
- Day
- Hour
- Month
- Month Name
- Year

Layout

Rows: Hour

Columns: Species

Measures: Value

Properties

Report Options...

Species	CO	FINE	NO2	O3	PM10	PM25
Hour	Value	Value	Value	Value	Value	Value
0	,4	9,2	31,9	25,1	16,3	
1	,4	8,7	28,	25,4	15,5	
2	,4	8,6	26,4	24,4	15,7	
3	,4	8,5	26,9	22,7	15,9	
4	,4	9,4	31,3	19,4	16,7	
5	,4	10,9	40,3	16,1	19,1	
6	,4	11,4	45,5	16,3	20,9	
7	,4	10,5	44,8	19,8	20,7	
8	,4	9,7	43,	24,9	20,3	
9	,4	9,3	40,6	30,4	19,1	
10	,4	8,7	39,6	34,4	18,4	
11	,4	8,6	39,8	38,	18,	
12	,3	8,6	39,6	41,	17,5	
13	,4	8,1	39,9	43,4	16,8	
14	,3	8,4	40,8	44,7	17,4	
15	,3	8,6	41,8	45,4	17,2	
16	,3	8,8	42,4	44,6	16,5	
17	,4	9,4	43,5	42,3	16,6	
18	,4	10,3	45,1	38,1	16,9	

Now we have the data in a table format we can present it in a graph.

- Click on the **Switch to chart format** pull down menu icon and select the **Line** chart.

Available fields (12) for: Air Quality Data

Find: View

Measure Value

Level (Names, Categories, etc.)

- Authority
- Location (2)
- Site (1)
- Site Type
- Species

Time

- Date
- Day
- Hour
- Month
- Month Name
- Year

Layout

Rows: Hour

Columns: Species

Measures: Value

Properties

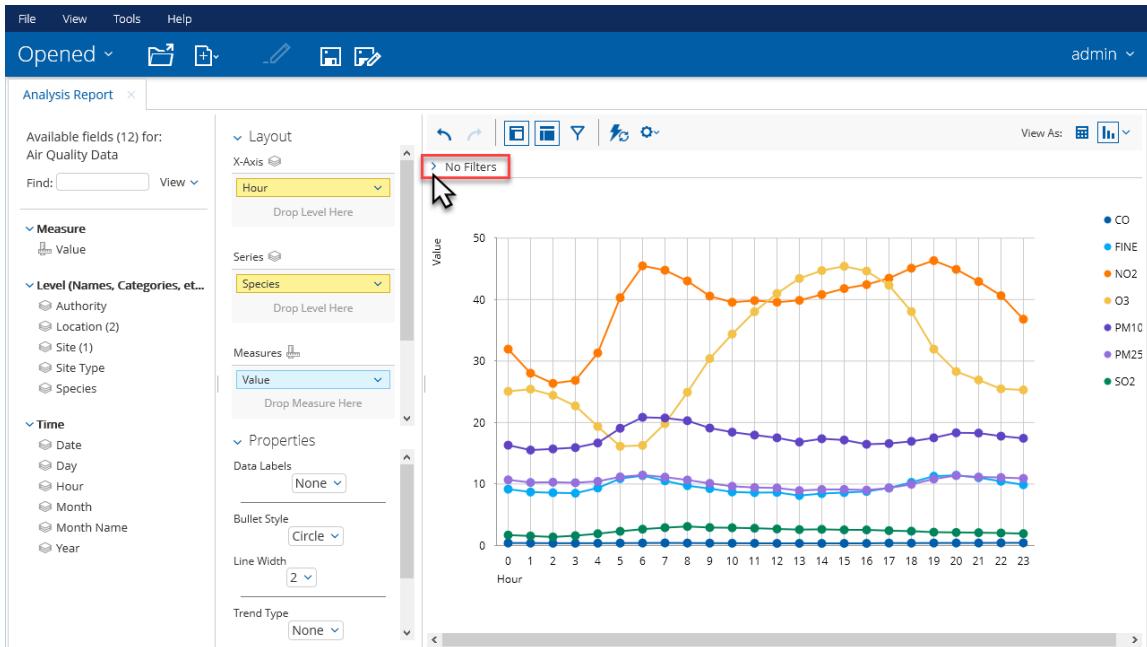
Report Options...

View As:

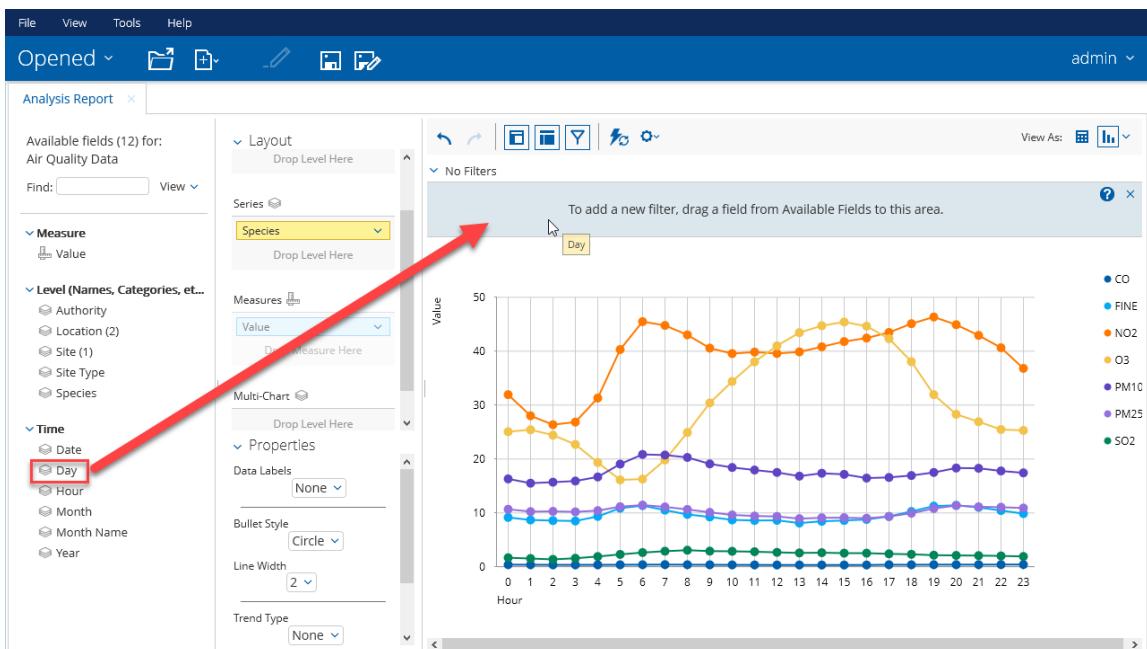
- Column
- Stacked Column
- 100% Stacked Column
- Column-Line Combo
- Bar
- Stacked Bar
- 100% Stacked Bar
- Line**
- Area
- Pie
- Sunburst
- Scatter
- Heat Grid
- Geo Map

Species	CO	FINE	NO2	O3	PM10	PM25
Hour	Value	Value	Value	Value	Value	Value
0	,4	9,2	31,9	25,1	16,3	
1	,4	8,7	28,	25,4	15,5	
2	,4	8,6	26,4	24,4	15,7	
3	,4	8,5	26,9	22,7	15,9	
4	,4	9,4	31,3	19,4	16,7	
5	,4	10,9	40,3	16,1	19,1	
6	,4	11,4	45,5	16,3	20,9	
7	,4	10,5	44,8	19,8		
8	,4	9,7	43,	24,9		
9	,4	9,3	40,6	30,4		
10	,4	8,7	39,6	34,4		
11	,4	8,6	39,8	38,		
12	,3	8,6	39,6	41,	17,5	
13	,4	8,1	39,9	43,4	16,8	
14	,3	8,4	40,8	44,7	17,4	
15	,3	8,6	41,8	45,4	17,2	
16	,3	8,8	42,4	44,6	16,5	
17	,4	9,4	43,5	42,3	16,6	
18	,4	10,3	45,1	38,1	16,9	

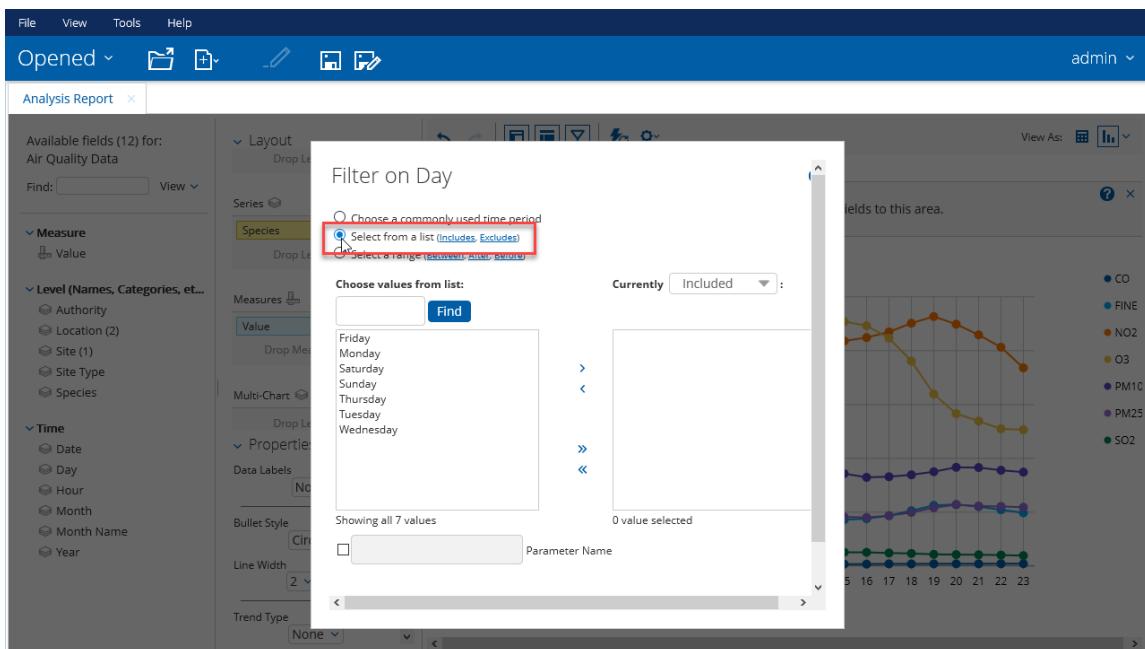
Your resulting chart should like this:



6. Click on No Filters to open the **filter area** (see screenshot above) and drag the Time.Day **attribute** onto the filter area.



7. Choose the option **Select from a list**



8. Move all day **values** from the left side area to the right side area by clicking on the double right arrow (**>>**) and specify a **Parameter Name**. In this case day (but can be any name).

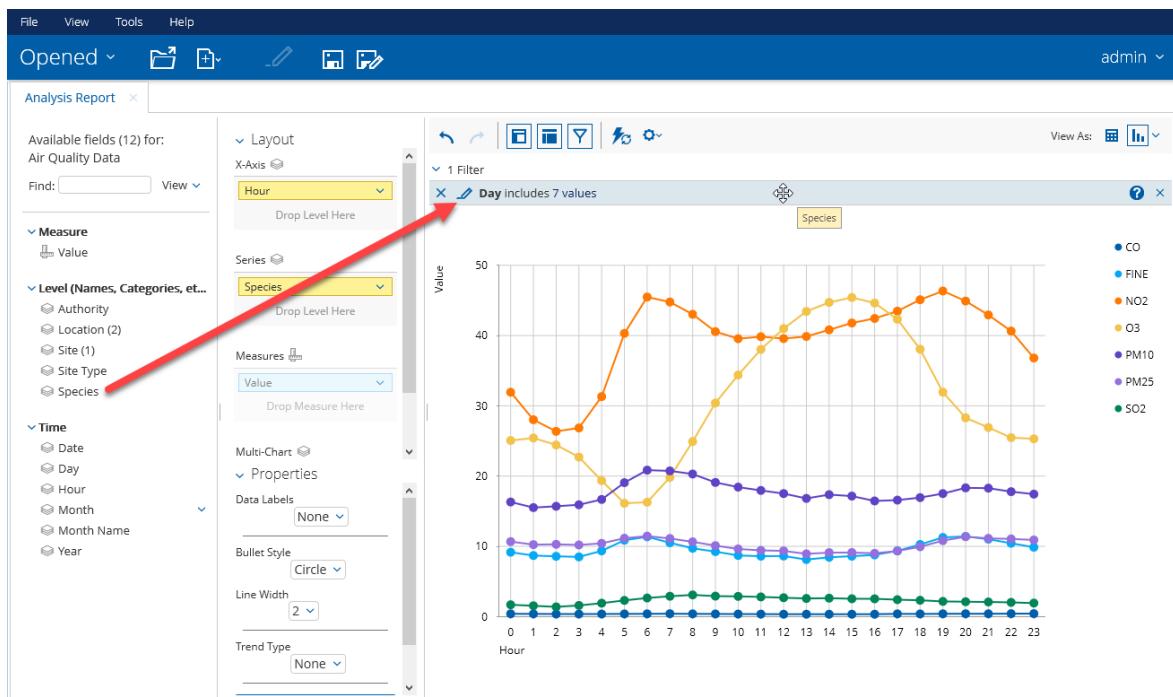


A filter selection can be reverted by selecting **Currently Excluded**.



Parameters specified in a filter can be used in dashboards to filter the analysis via Prompts.

9. Press **OK** to save the filter and close the window
10. Do the same with the Species **attribute**. Drag it onto the filter area.



11. Move all Species **values** from the left side area to the right side area by clicking on the right-pointing double angle quotation mark (») and specify species as the **Parameter Name**

Filter on Species

Select from a list ([Includes](#), [Excludes](#))
 Match a specific string ([Contains](#), [Doesn't Contain](#))

Choose values from list: Currently Included :

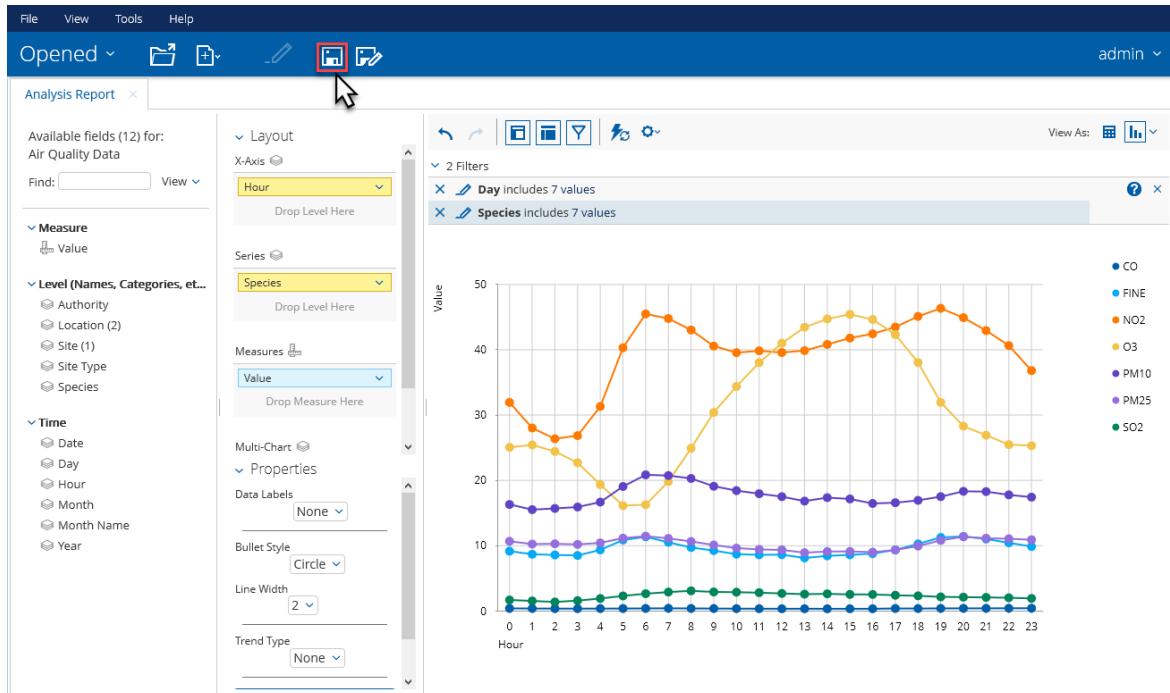
<input type="button" value="Find"/>	>	<	»	«
CO FINE NO2 O3 PM10 PM25 SO2	> CO FINE NO2 O3 PM10 PM25 SO2	< CO FINE NO2 O3 PM10 PM25 SO2	» CO FINE NO2 O3 PM10 PM25 SO2	« CO FINE NO2 O3 PM10 PM25 SO2

Showing all 7 values 7 values selected

species Parameter Name

12. Press **OK** to save the filter and close the window

Your analysis should look like this:



13. Click on the save icon to save the analysis

14. Enter Average Reading by Hour as the name you save the analysis under and **Press Save**. The definition of the analysis is now saved in the central Pentaho Repository.

Save

Filename:

Location:

Name	Type	Date Modified

Save **Cancel**

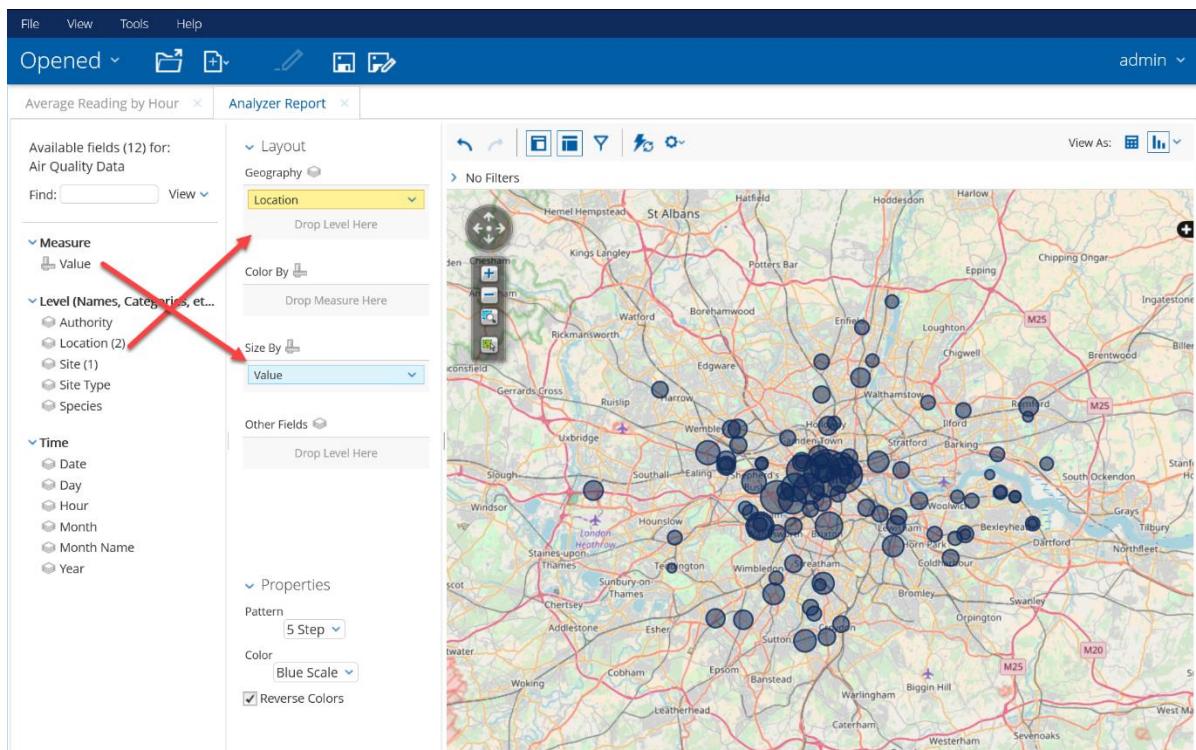
Creating the Geospatial Report

Before we continue with the dashboard we are going to create another Analysis: the Geospatial Report. We will do this again on the [Air Quality Data](#) data model.

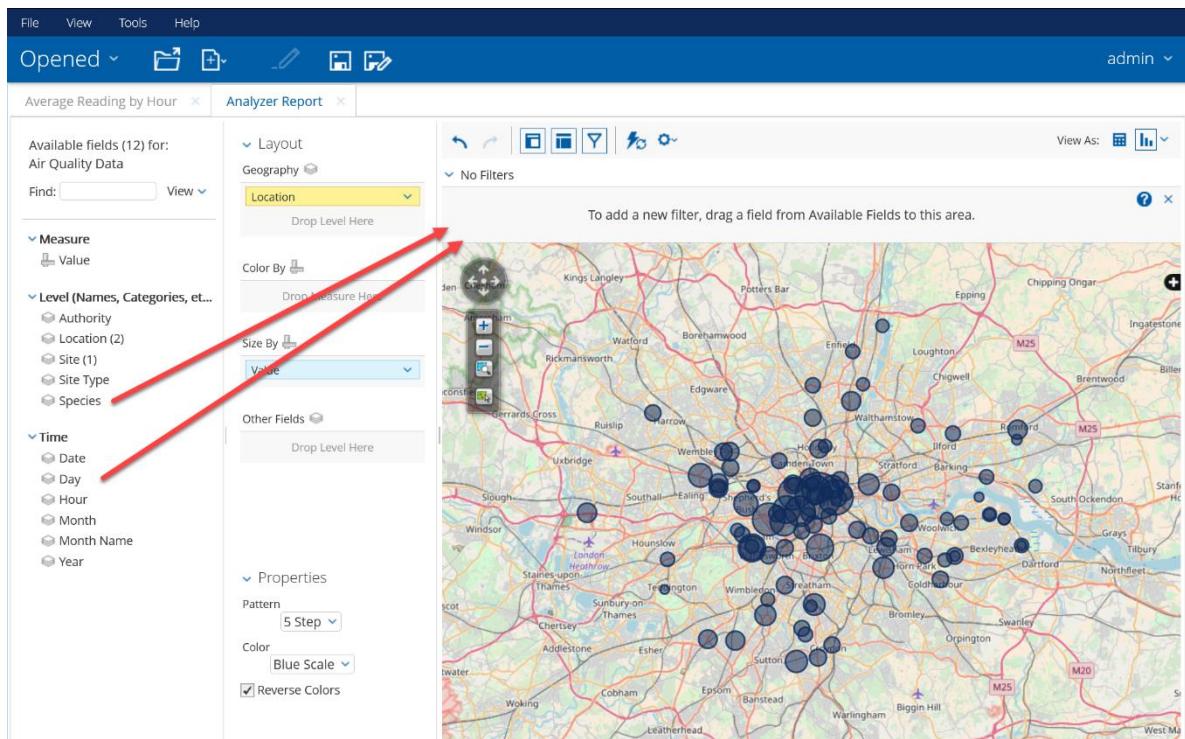
1. First switch to **Chart view** ([View As:](#)) to select the Geo Map

The screenshot shows the Pentaho Analyzer Report interface. The top navigation bar includes File, View, Tools, Help, and a user dropdown. Below the navigation is a toolbar with Opened, View, Tools, Help, and a user dropdown. The main workspace displays an 'Analyzer Report' titled 'Average Reading by Hour'. On the left, a sidebar lists available fields: Air Quality Data, Measure (Value), Level (Names, Categories, etc.), Location (2), Site (1), Site Type, Species, Time (Date, Day, Hour, Month, Month Name, Year), and Properties. The central area shows a 'Table' layout with 'Rows' and 'Columns' sections. A sidebar on the right, titled 'View As:', lists various chart types: Column, Stacked Column, 100% Stacked Column, Column-Line Combo, Bar, Stacked Bar, 100% Stacked Bar, Line, Area, Pie, Sunburst, Scatter, Heat Grid, and Geo Map. The 'Geo Map' option is highlighted with a red box. A tooltip at the bottom of the sidebar says 'Drag an available field to the required layout zone' and '(Marked with a red asterisk in the layout panel.)'. At the bottom right, there are 'Need help getting started?' and 'Help' buttons.

2. Drag the Location **attribute** to the Geography and Value **metric** to the Size **Layout area**

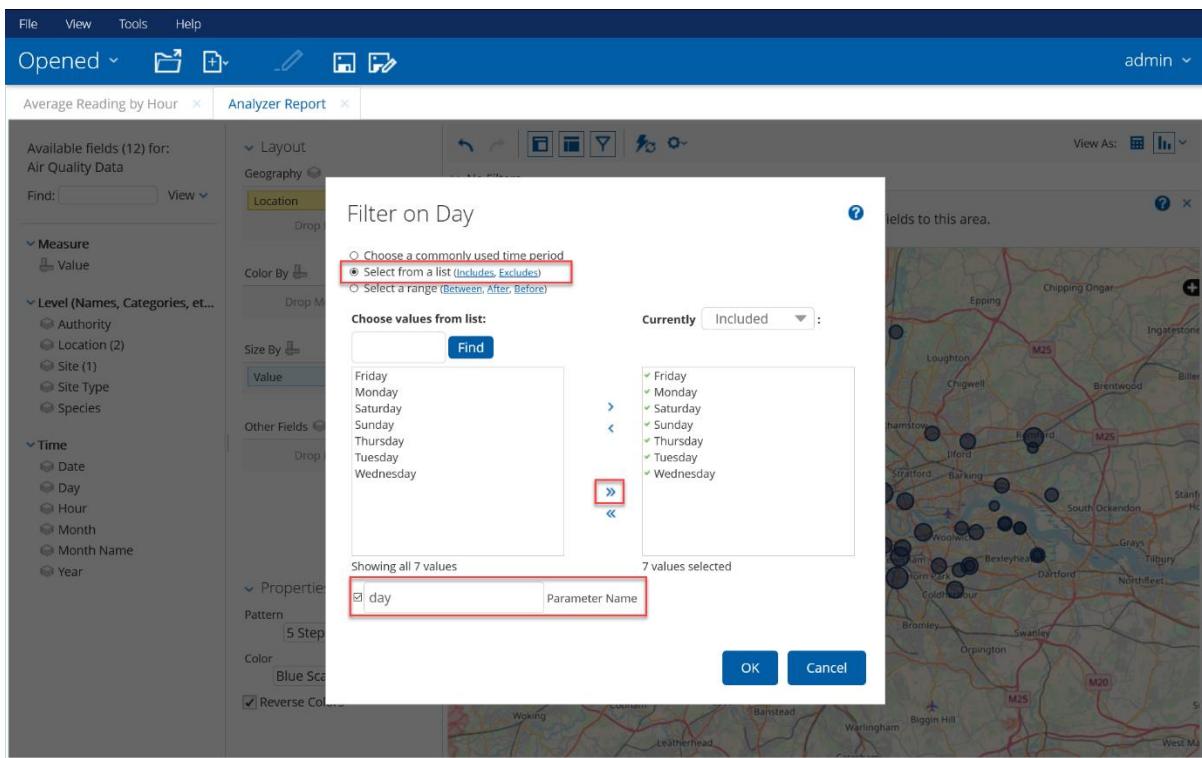


3. Drag the Species and Day attributes to the filter area.

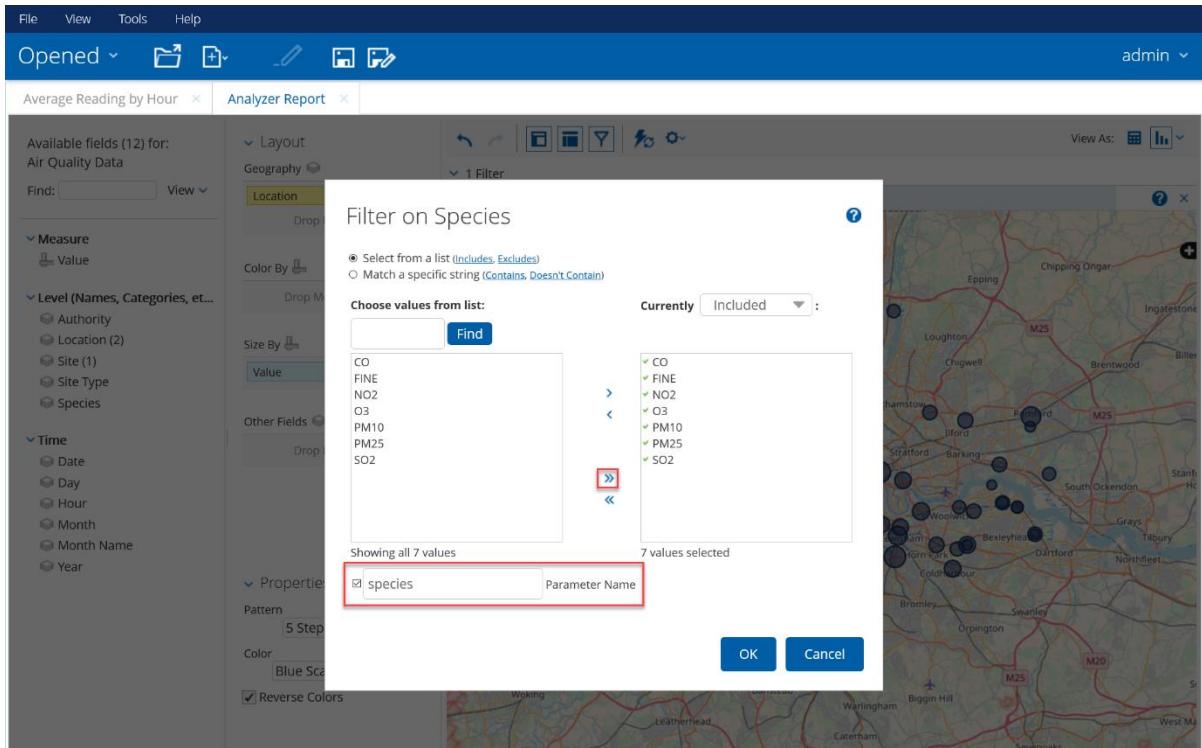


4. Configure the Day **filter** as shown below. Make sure you will specify a **Parameter Name**.

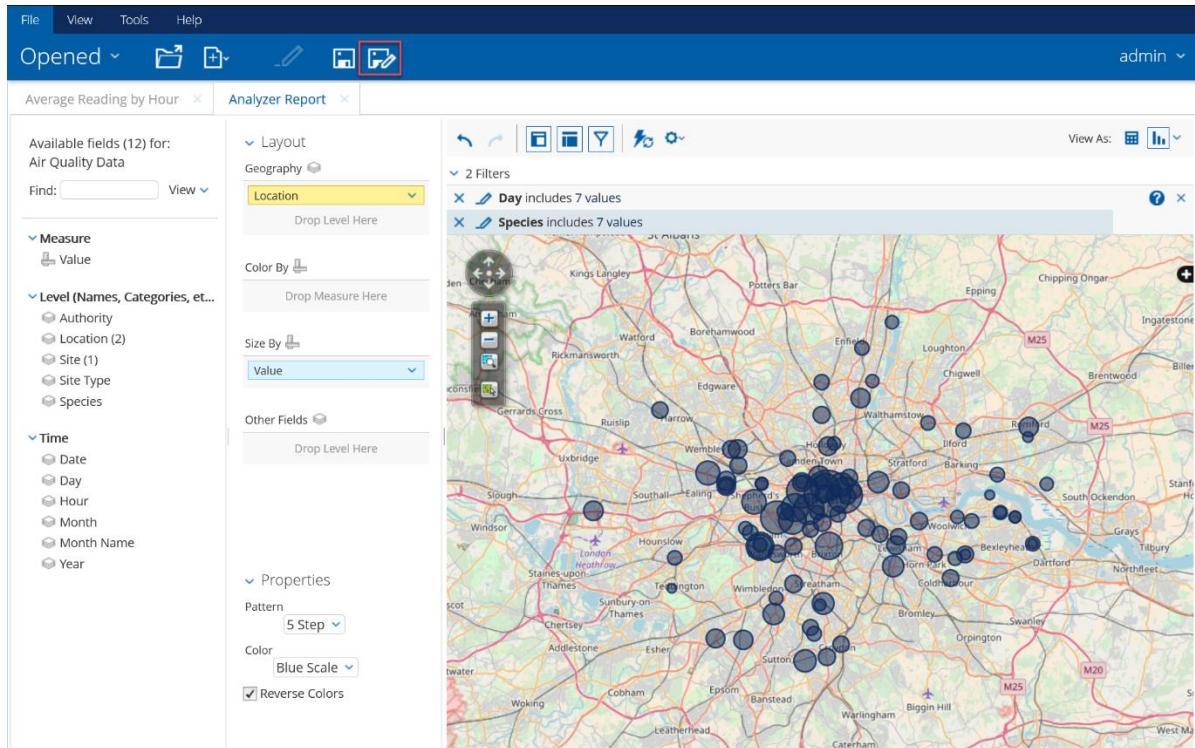
Pentaho LAQ Handbook



5. Configure the Species **filter** as shown below. Make sure you will specify a **Parameter Name**.



6. The result of the geomap should be like shown below:



7. Now save the geomap by clicking on the **Save** icon (). Enter Average Reading by Site - Geo as the **Filename** for the geomap. Press **Save** to save and close the window.

Save As

Filename:

Location:

Name	Type	Date Modified
Average Reading by Hour	File	Sep 14, 2017 1:10:40 PM

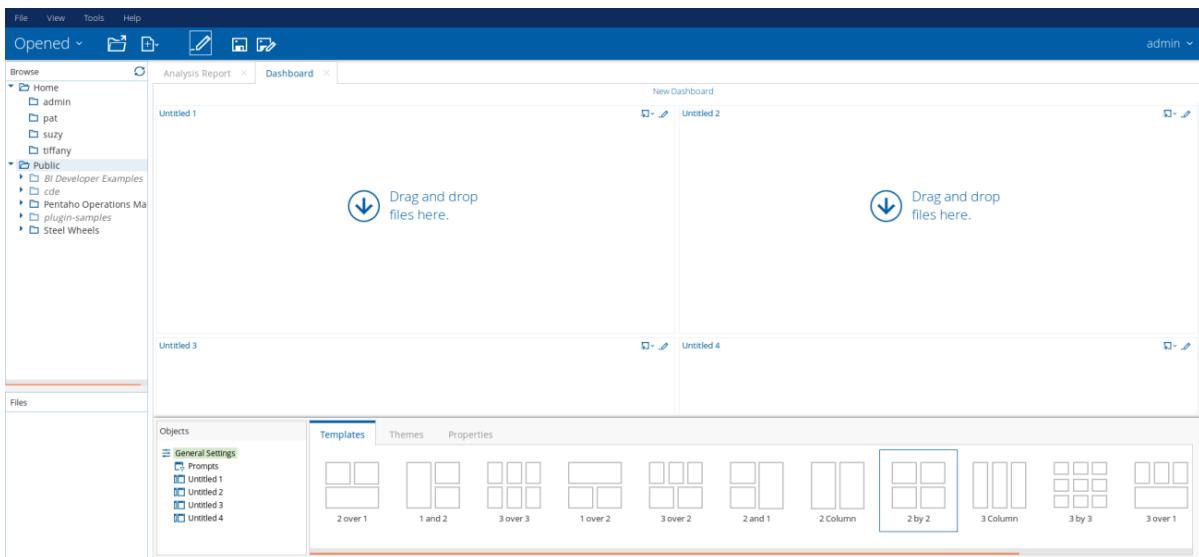
Save **Cancel**

Creating the Air Quality Dashboard

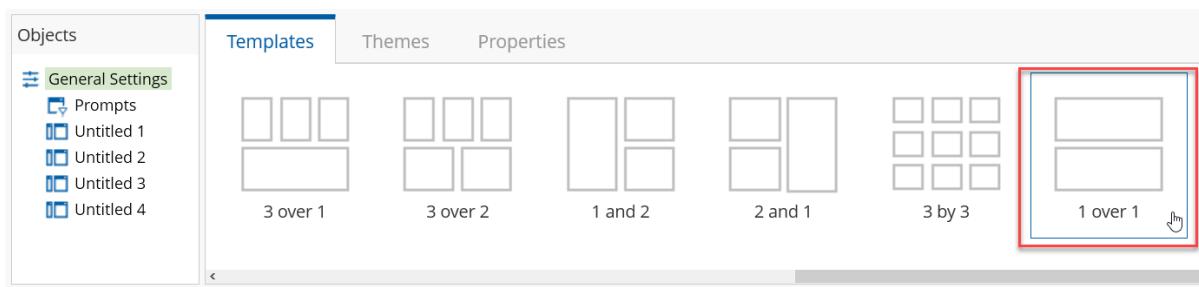
In the Pentaho User Console, you also can create interactive and embeddable dashboards. Now let's create the Air Quality Dashboard. This dashboard will contain the two previously defined analytics:

- Average Reading by Hour
- Average Reading by Site - Geo

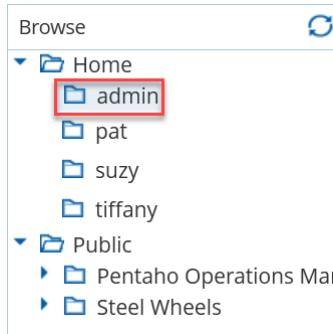
1. In the Pentaho User Console (<http://localhost:8080>), click on the **Create New > Dashboard** buttons. The Dashboard Designer should popup.



2. Choose the 1 over 1 **template** for the dashboard



3. Open the Home > admin **folder** in the **Browse** pane



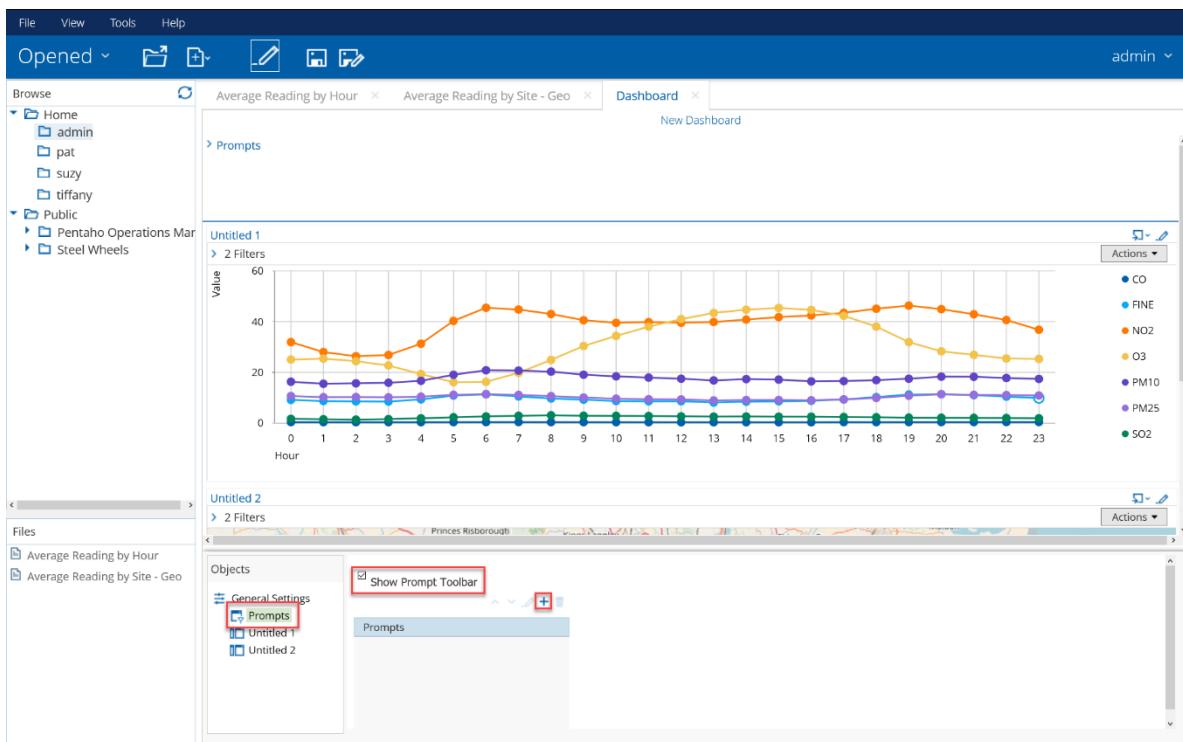
 The Browse pane shows all objects stored in the central Pentaho Repository, available for the dashboard. Job and transformation will not be seen here.

4. Drag the following predefined analysis:

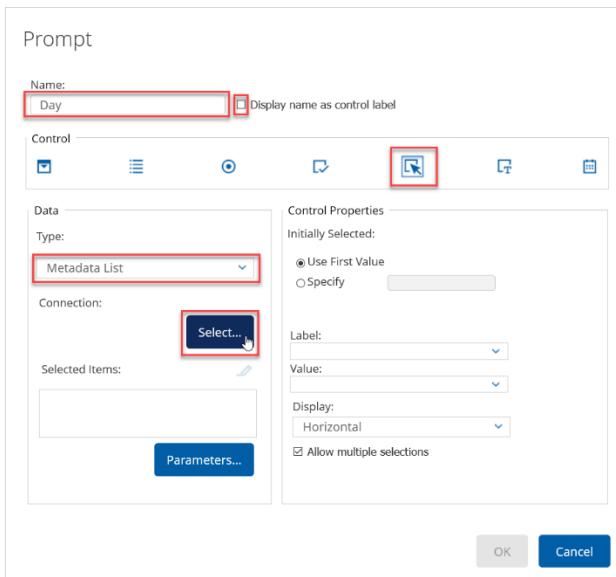
- Average Reading by Hour onto the top pane of the dashboard
- Average Reading by Site - Geo onto the bottom pane of the dashboard

The screenshot shows the Pentaho dashboard editor. At the top, there's a toolbar with various icons. Below it, the title bar shows 'Opened' and the current dashboard name 'Average Reading by Hour'. The main area has a sidebar on the left labeled 'Browse' with the same repository structure as the previous screenshot. On the right is the dashboard canvas with a placeholder 'Untitled 1'. In the bottom left corner, there's a 'Files' pane containing two items: 'Average Reading by Hour' and 'Average Reading by Site - Geo'. Red arrows point from these items towards the dashboard canvas. In the bottom center, there's an 'Objects' pane with tabs for 'General Settings', 'Prompts', 'Untitled 1', and 'Untitled 2'. A blue arrow points from the 'Prompts' tab towards the dashboard canvas. To the right of the objects pane, there are several layout templates shown as grid icons: '3 over 1', '3 over 2', '1 and 2', '2 and 1', '3 by 3', '1 over 1', and '3 over 3'. A tooltip 'Drag and drop files here.' with a downward arrow is positioned near the top of the dashboard canvas.

5. Since the analysis are in place now we can start filtering the data on Day and Species. This needs to be handled by prompts. Click on **Prompts** in the **Objects** pane and activate **Show Prompt Toolbar**



- Click in the blue plus (+) icon to add a prompt for Day and configure the properties marked with red boxes in the screenshot below.



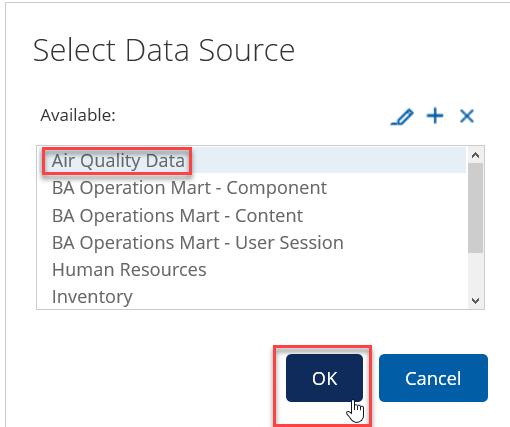
Press **Select...** to define the query for the metadata list.

 The values for the prompt can be created in three ways:

- 1) Static List, where you must define your own list with labels and values
- 2) SQL List: a SQL Query on a data source. Make sure you know the table- and column names, that you specify a distinct list and sort it (order by)

3) Metadata List: a query to retrieve a list of display names and corresponding values from a metadata data source

Select Air Quality Data as the **Data Source** for your query.



- In the **Query Editor** drag the Daystring and Weeknumday attributes on the **Selected Columns** area. Drag the Weeknumday attribute on the **Order By** area. Specify the ascending **Order**.

Selected Columns:	
Aggregation	Column
NONE	Daystring
NONE	Weekdaynum

Combine	Aggregation	Column	Comparison	Value	Default

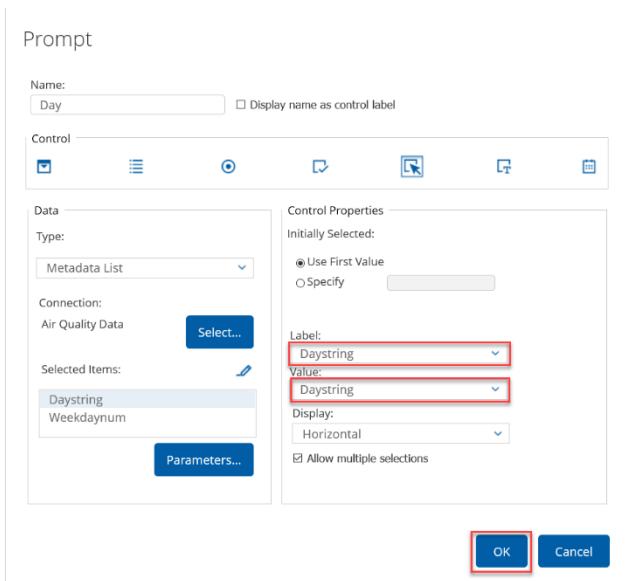
Order By:		
Aggregation	Column	Order
NONE	Weekdaynum	ASC

Row Limit (must be greater than or equal to 0):

Preview OK (highlighted with a red box) Cancel

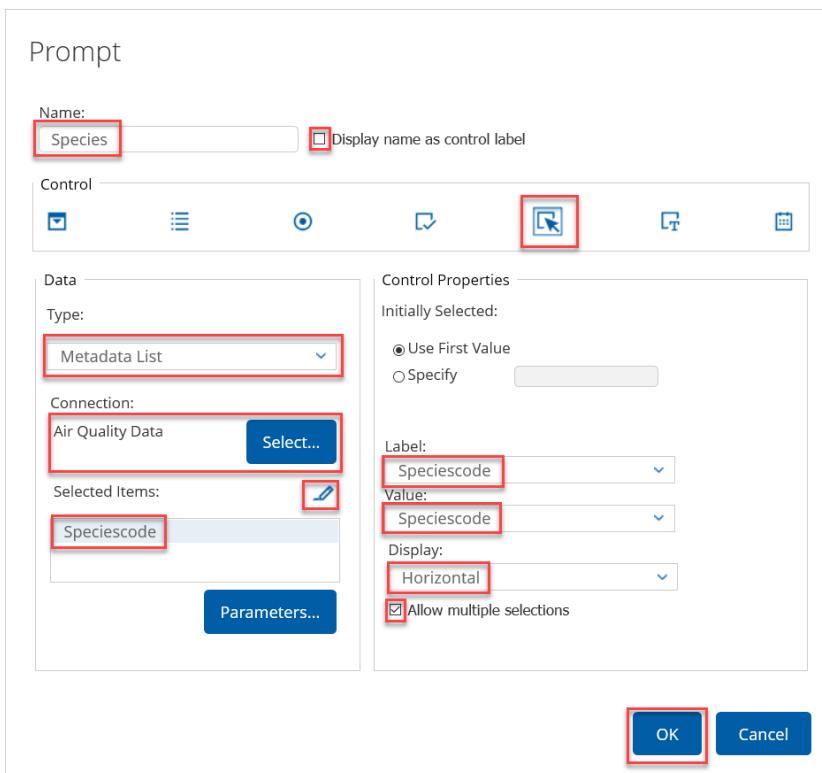
Press **OK** to save and close the window

8. Coming back into the **Prompt** window select Day for the **Label** and **Value**



Press **OK** to save and close the Prompt window

9. Click in the blue plus (+) icon to add a prompt for Species and configure the properties marked with red boxes in the screenshot below. DO NOT press OK.



Press **Select...** to define the query for the metadata list.

10. In the **Query Editor** add the Speciescode attributes to the **Selected Columns** and the **Order By** area by using the blue greater signs (>). Specify the ascending **Order**.

Query Editor

Categories / Columns

- Aqd
 - Authorityname
 - Day
 - Dayofweek
 - Hour
 - Lat
 - Lon
 - Measurementdate
 - Month
 - Shortpostcode
 - Sitecode
 - Sitename
 - Sitetype
 - Speciescode**

Selected Columns:

Aggregation	Column
NONE	Speciescode

Conditions:

Combine	Aggregation	Column	Comparison	Value	Default

Order By:

Aggregation	Column	Order
NONE	Speciescode	ASC

Row Limit (must be greater than or equal to 0):

Preview **OK** Cancel

Press **OK** to save and close the window

11. Press **OK** to close the Prompt window
12. Select the first Untitled object (this is the Average Reading by Hour analysis) and give it the title Average Hourly Reading on.



Tip: you can copy the Content name shown in the Objects panel

Select the correct **Source** for the **Parameters** as shown in the screenshot below. Click on the **Add parameters to title**

Objects

- General Settings
- Prompts
- Untitled 1**
- Untitled 2

Title: **Average Hourly Reading on {day}**

Refresh Interval (sec)

Content:

Average Reading by Hour

Parameters

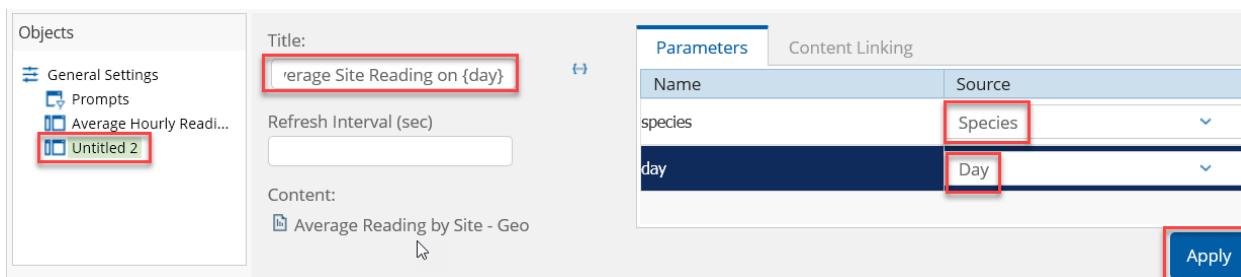
Name	Source
species	Species
day	Day

Content Linking

Apply

Press **Apply** to save

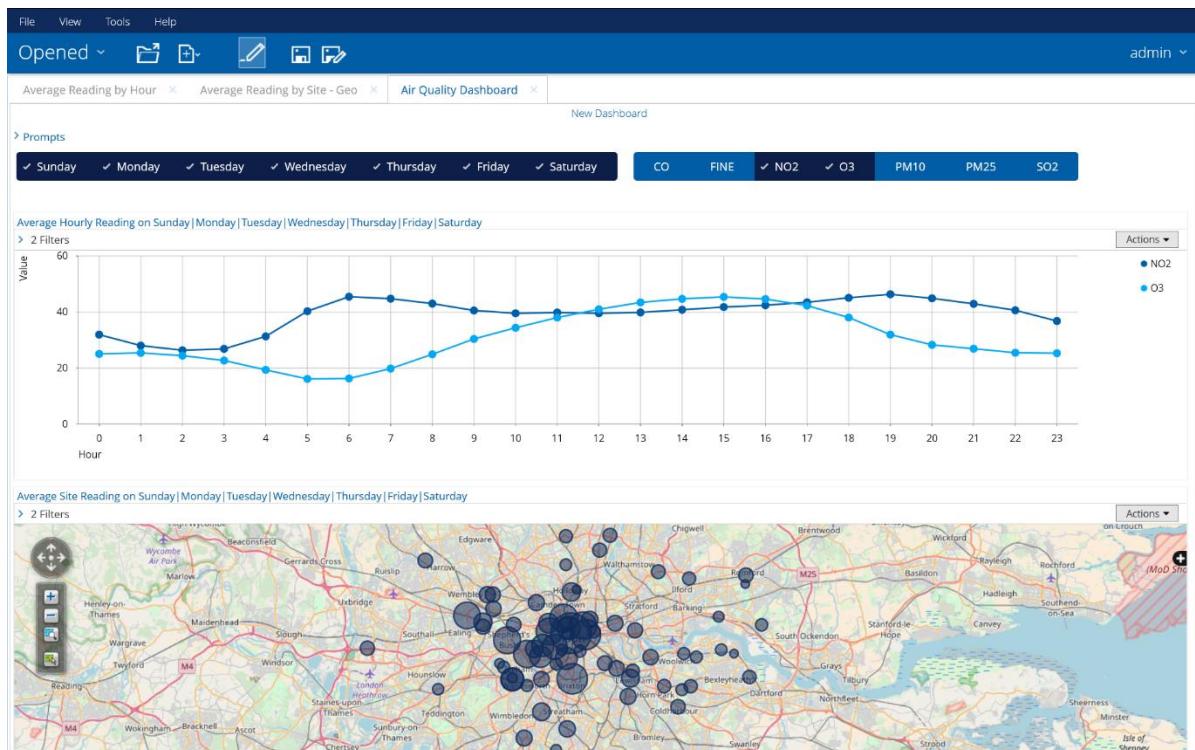
13. Select the second Untitled object (this is the Average Reading by Site - Geo analysis) and give it the title Average Site Reading on. Select the correct **Source** for the **Parameters** as shown in the screenshot below. Click on the **Add parameters to title**  icon to add the parameters to the title.



Name	Source
species	Species
day	Day

Press **Apply** to save

14. The result for the dashboard should look like this:



15. Click on the **Save As**  icon to save the dashboard as Air Quality Dashboard