

Power BI Notes

Dylan Johnston

Notes based on the Udemy course by Phillip Burton

<https://www.udemy.com/course/70-778-analyzing-and-visualizing-data-with-power-bi/>



Contents

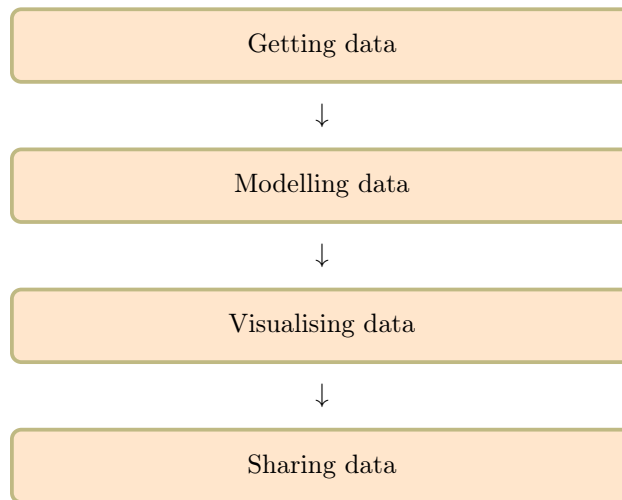
0	Introduction	3
0.1	The four stages of a Power BI workflow	3
0.2	Dummy data (for examples throughout)	4
1	Visualising Data	5
1.1	Creating a visualisation	5
1.2	Formatting a visualisation	6
1.3	Matrices, bar charts, and interaction between visuals	6
1.4	Add more to visualisations (filters, sorts, slicers, ...)	7
1.5	Other visualisations	10
1.6	Maps	10
1.7	Measuring performance (KPIs, gauges, and cards)	11
1.8	Identify patterns & trends	12
2	Getting Data (PowerQuery)	13
2.1	Introduction and Home tab	13
2.2	Getting multiple files	14
2.3	Transform tab	14
2.4	Text and Numbers	15
2.5	Dates and Time	16
2.6	Adding Columns	16
2.7	Converting text to number (dealing with locale)	17
2.8	View and Help tabs	17
2.9	Importing from SQL Server	17
3	Modelling Data	18
3.1	Introduction to models	18
3.2	Logical DAX functions	18
3.3	Stats and measures	19
3.4	Mathematical Functions	20
3.5	Text Functions	20
3.6	Look Up Value function	21
3.7	Filter and Value functions	21
3.8	Date Intelligence functions	22
3.9	Calculated Tables & Visuals	22
4	Sharing Data (The Power BI Service)	23
4.1	Row level security	23
4.2	Dashboards	23
4.3	Managing Semantic models	24
4.4	Managing workspaces	24

Chapter 0

Introduction

0.1 The four stages of a Power BI workflow

There are roughly four stages in a Power BI workflow.

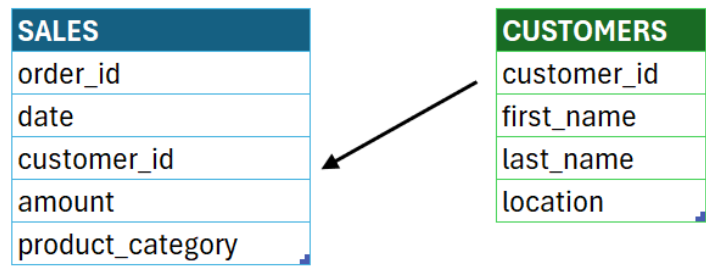


The Udeemy course, and therefore these notes, follow the order:

- Visualising data
- Getting data
- Modelling data
- Sharing data

0.2 Dummy data (for examples throughout)

It will be useful to give small examples throughout (and for this, some dummy data is needed). A Sales table and a Customers table will be used, related by a customer_id, as follows:



Sales

order_id	date	customer_id	amount	product_category
1	03/10/2024	C001	454	Sports
2	03/10/2024	C003	388	Electronics
3	22/10/2024	C003	259	Electronics
4	03/11/2024	C002	283	Home
5	05/11/2024	C001	322	Sports
6	12/12/2024	C001	549	Electronics
7	12/12/2024	C002	401	Home
8	12/12/2024	C003	450	Sports
9	09/01/2025	C001	143	Electronics
10	28/02/2025	C002	555	Home

Customers

customer_id	first_name	last_name	location
C001	Alice	Anderson	London
C002	Bob	Bennett	Manchester
C003	Charlie	Clarke	Newcastle

Chapter 1

Visualising Data

This chapter looks at the creation and formatting of visualisations in Power BI, covering the different types of charts, interactive elements like slicers and filters, and tools for measuring performance.

1.1 Creating a visualisation

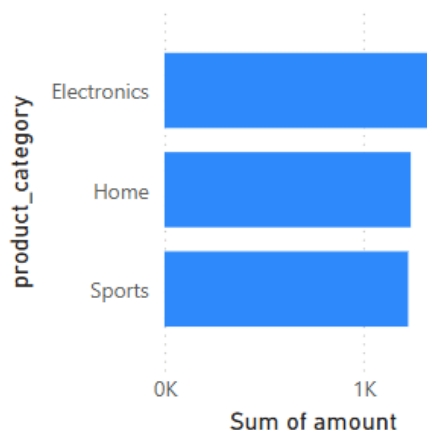
To quickly import data, go to **Home** → **Get data**.

For example, you can import from an Excel Workbook. Selecting a workbook opens a navigator window showing all sheets and named ranges from the workbook. Select whichever tables you wish to import, then press “Load”.

Any imported data will appear in the Data pane.

Checking some of the fields within the imported data produces a visual. For example, checking fields ‘product_category’ and ‘Σ amount’ produces the following visual:

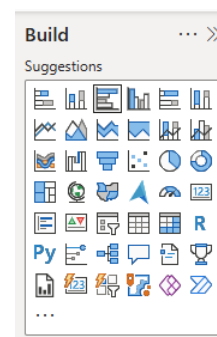
Sum of amount by product_category



By default, numerical fields are summed (e.g. ‘Σ amount’). To change this, select the field from the Data pane, navigate to **Column tools** → **Properties** → **Summarisation** and select a different aggregation function.

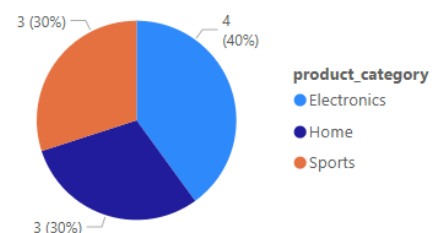
Also observe the **Formatting** and **Structure** area, where the data type and formatting of the field can be changed (e.g. to a currency, percentage).

To change a visualisation type (e.g. from a bar chart to a pie chart), click into the visualisation and open the Build Pane.



For example, clicking on pie chart (row 3, column 5) converts the bar chart into the following (note also that the numerical column ‘Σ amount’ has been changed to count instead of sum).

Count of amount by product_category



1.2 Formatting a visualisation

Visuals are formatted using the Format pane. If this is hidden, click on a visual, navigate to **View** → **Show panes** → **Format**.

From this pane, changes can be made to the title, colours, legend, etc.

Format Mode

Double-clicking on a visual will enclose it with square selection handles (rather than circles). This is “Format Mode”. Right-clicking on elements within the visualisation will jump to the relevant format options within the Format pane.

Moving visualisations

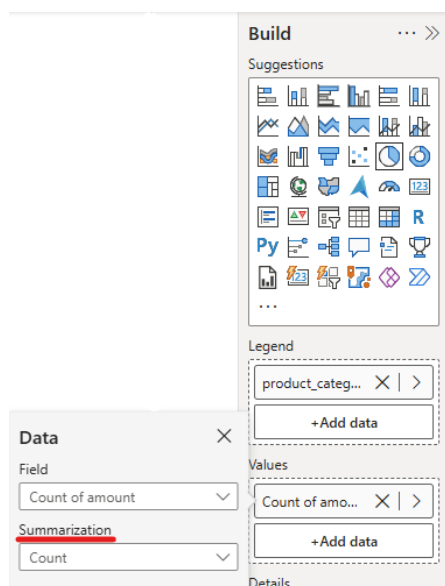
Clicking and holding on some white space within a visualisation will allow you to move it around. Visualisations can also be moved (precisely) within the “Size and style” section of the Format pane.

Format painter

To copy the formatting from one visualisation for use in another you can use the Format painter. This is found at **Home** → **Clipboard** → **Format Painter**.

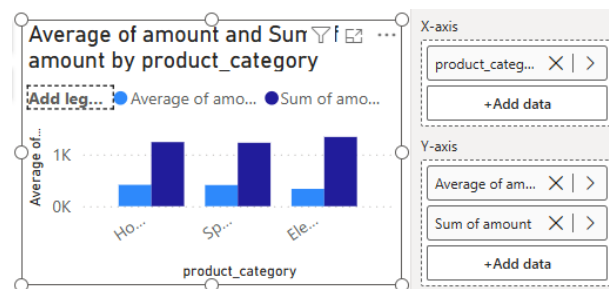
Changing aggregate functions locally

To change how a numerical column is aggregated locally (i.e. only within a specific visualisation), select the visualisation and adjust the function in the “Values”/“Columns” section of the Build pane.



Dragging fields into Build pane

Fields can be dragged and dropped into the Build pane of a visualisation in order to add them to the visual. For example, you can drag ‘ Σ Amount’ into the Y-axis section, change the aggregate function to sum, (and also change pie chart to clustered column chart,) and obtain the following:



Right-clicking on a field within the Build Pane allows you to rename it locally (i.e. give it a new name only for the current visual).

1.3 Matrices, bar charts, and interaction between visuals

• Matrix

This is analogous to an Excel pivot table. After adding a matrix (via navigating to **Home** → **Insert** → **Matrix**), you will see “Rows”, “Columns” and “Values” sections within the Build pane.

For example, dragging ‘customer_id’ into “Rows”, ‘product_category’ into “Columns” and ‘ Σ Amount’ into “Values” produces:

customer_id	Electronics	Home	Sports	Total
C001	692		776	1468
C002		1239		1239
C003	647		450	1097
Total	1339	1239	1226	3804

Toggling the “Data point table” feature, found at **Data/Drill** → **Show** → **Data point table**, then clicking on a value cell in a matrix visual will open a new page displaying the rows of the original data contributing to the clicked value.

• Bar/Column chart variants

There are many variants of the bar/column chart, including clustered, stacked, and 100% stacked.

Interactions between visualisations

Clicking on a visualisation and navigating to **Format** → **Interactions** → **Edit interactions** will allow you to change how visualisations interact with each other.

After toggling on “Edit interactions”, all other visualisations will have the following icons in the top right corner:



Respectively, these icons enable filtering (i.e. showing only the selected data), highlighting (i.e. highlighting the selected data) or disable interaction completely.

1.4 Add more to visualisations (filters, sorts, slicers, ...)

The following two items are a great way to add extra detail to a visualisation, or to point the reader to something of particular interest.

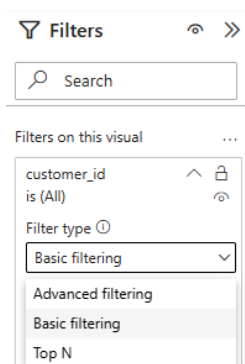
- Text Box
Home → **Insert** → **Text Box**
Great for giving extra context or information about a visualisation.
- Images and Shapes
Insert → **Elements** → ...

Filters

Filters can be defined at the visualisation level, page level or report level (so, for example, you can filter different visuals by different things).

The Filters pane is enabled via navigating to **View** → **Show panes** → **Filters**.

Selecting a visual will allow you to set filters. For example, selecting a matrix visual gives the following filtering options:



Basic filtering allows you to manually select which values to include. This is good for discrete values (e.g. names) but not usually for ranges (e.g. dates or numerical values), as selecting the desired values manually would be time-consuming and not very dynamic.

Advanced filtering allows you to filter based on conditionals. For text, this includes “contains”, “starts with”, “is blank”, and their negations. For numerical and date data, there are options such as “less than”, “greater than”, “is” etc.

Top N filtering only shows the top (or bottom) *N* items, where the data is ranked based on a specified field (e.g. alphabetical, or largest numerically). Note that Top N filtering is only available on the level of visualisations.

Slicers

A slicer is essentially an interactive filter. It is a type of visualisation.

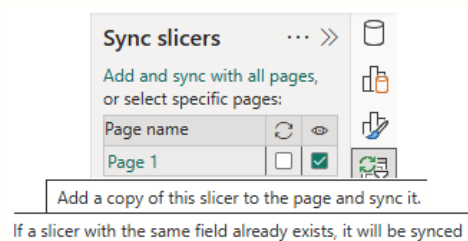
To add a slicer, go to **Home** → **Insert** → **Slicer**



Add fields to the slicer by dragging them from the Data pane into the “Field” section of the Build pane.

Use the Format pane to change font sizes, colours, add a title, change alignment, enable/disable multi-select (for discrete data) and more.

By default, slicers are at the page level. This can be changed to affect the whole report via **View** → **Show panes** → **Sync slicers**



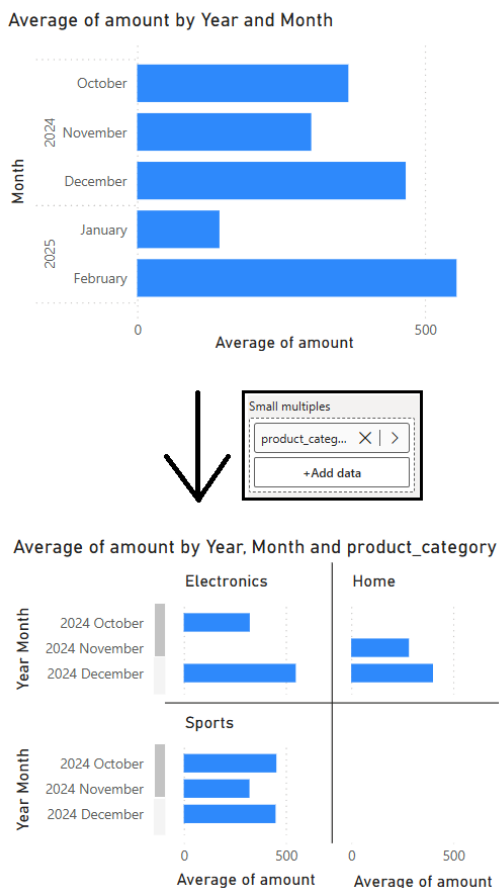
Sorting

To sort rows in a table (visualisation), click on the up/down arrow in the column header. To sort data within a general visualisation, click the ellipsis/more options icon (\cdots) and select the desired sort by option, as well as whether the sort should be ascending or descending.

You can sort the values in a field based on the corresponding values in another field (similar to using a key in a Python sort). To do this, select the field you want to change the sort order of, then navigate to **Column tools** → **Sort** → **Sort by column**.

Small multiples

For some visualisations (such as bar/column/line visuals), you can break a large visual into multiple smaller visuals, each displaying data for a single value in a desired field. To achieve this, drag the desired field into the "Small Multiples" section of the Build pane.



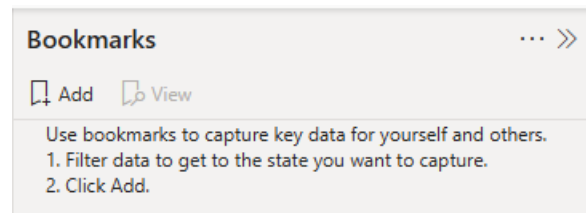
To change the order in which the small visuals are displayed, click the more options icon (\cdots) and use "Sort small multiples".

Bookmarks

Bookmarks store the active page, along with any applied filters, slicers, and sort orders.

They are useful for presentations, as they remove the need to adjust sorts and filters live.

View → **Show panes** → **Bookmarks**



The Selection Pane

This pane is enabled (as usual) by navigating to

View → **Show panes** → **Selection**

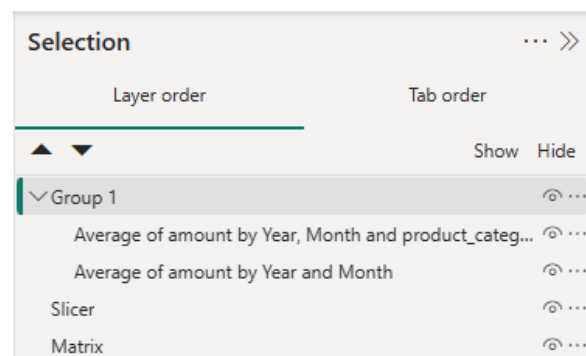
This pane can be used to show or hide visualisations on a page (e.g. if you want to hide a slicer, but still have it work in the background).

Layer order determines which visuals are on top if there are any overlaps.

Tab order determines the order in which the visuals will be cycled through when using the Tab key.

Selecting multiple visuals and clicking "More Options" (\cdots) allows you to group them. Essentially, these grouped visualisations are treated as one visualisation, affecting selection, visibility, sizing etc.

Groups can be ungrouped via the "More options" icon.



Drill through

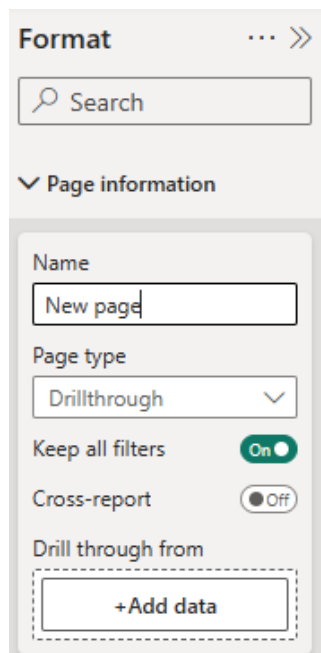
The drill through feature allows you to open a different page with filtered information based on a selected data point. It can be toggled on via

Data/Drill → Drill actions → Drill through.

To use the drill-through functionality, first you need to create the target page.

Then, in the Format pane, set the page type of this new target page to Drillthrough.

Finally, drag the desired field into the “Drill through from” section. This makes it so that the Drillthrough page is filtered based on the value chosen from the desired field.



Buttons

Buttons can be added to the report via

Insert → Elements → Buttons

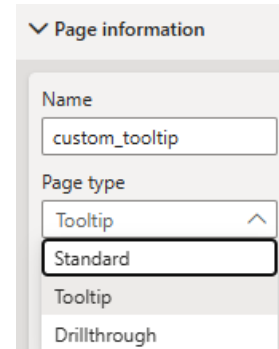
The Format pane allows you to set the action carried out.

Some options for the action include Back, Bookmark, Drill through, Page navigation, Web URL, Apply slicer

There are also sections within the Format pane to change the style (change the shape, add text, add a border, add shadows), rotate it, add a title, resize the button.

Tooltip pages

You can create pages that display as tooltips when hovering over various data points. This is done via setting up a new page and changing the page type from Standard to Tooltip.



Whilst not strictly necessary, it is also advisable to keep the page size small (this option is found in the “Canvas settings” section). Usually, size tooltip (320px × 240px) is good.

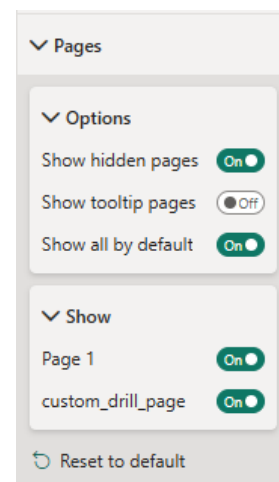
To set the tooltip used for each visualisation (on the original page), go to a visualisation’s Format pane, and modify the Tooltips section from within the Properties tab.

Navigators

Page and bookmark navigators can be added to a page via

Insert → Elements → Buttons → Navigator

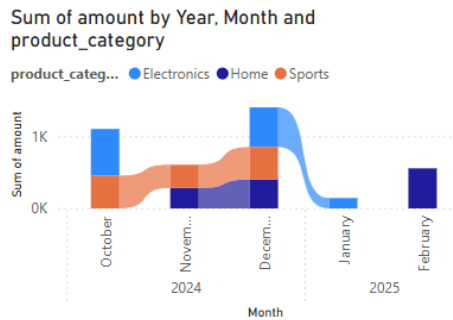
With a page navigator, you can enable/disable providing navigation to hidden pages, tooltip pages, and drill through pages from within the “Pages” section of the Format pane.



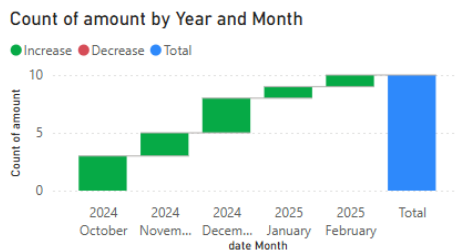
1.5 Other visualisations

Some other useful chart options are:

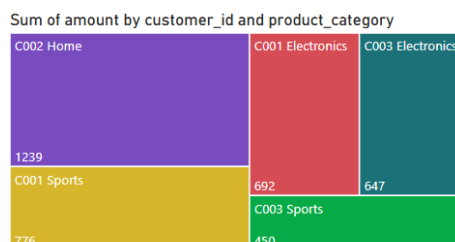
- Ribbon chart - a variation of the stacked column chart that highlights changes in ranking over time. It connects columns with ribbons, showing how values from the legend change in ranking over time.



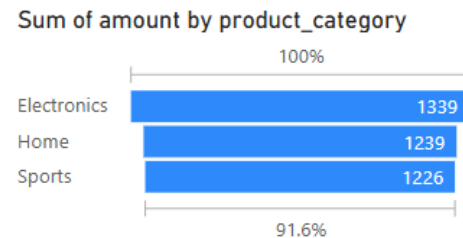
- Waterfall charts - used to visualise incremental changes in a value over time or across categories. A “Breakdown” section is present within the Build pane, that will show how each item in an added field contributes to the running total.



- Pie charts - a chart displaying the proportions of values in a field added to the “Values” sections. Different coloured slices of the pie are created by adding fields to the “Legend” section of the Build pane. Each slice can be further broken up by adding a field to the “Details” section of the Build pane.
- Treemaps - each value in the field(s) added to the “Category” section is represented by a rectangle whose area is proportional to its value with respect to the field in the “Values” section.



- Scatter charts and bubble charts (the latter is obtained by adding a field to the “Size” section of the Build pane for a scatter chart).
- Funnel charts - a chart that displays percentages of a total using centered horizontal bars. Note that when the data is ordered from largest to smallest, the chart takes on a funnel-like shape, hence the name.



1.6 Maps

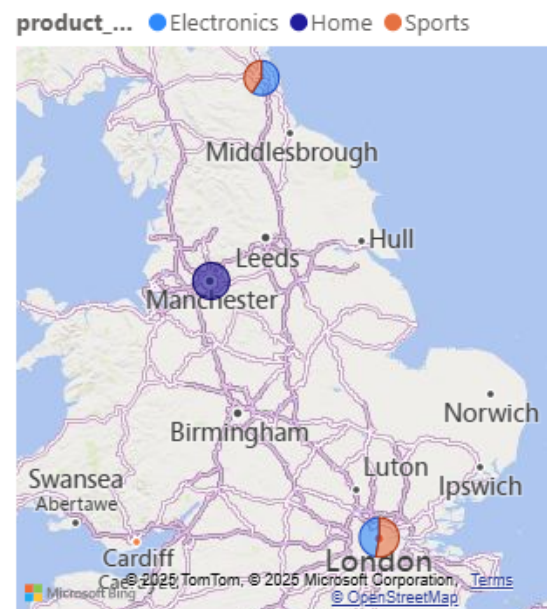
Maps can be added as visualisations

Insert → **Visuals** → **Maps** → **Map**.

Drag a field containing locations to the “Location” section of the Build pane. The bubbles appearing at each location can be scaled according to a numerical field by adding it to the “Bubble size” section

Adding a field to the “Legend” section will convert the bubbles to pies, showing proportions with respect to that field.

Sum of amount by location and product_category



Data categories

To change the data category of a field, select it within the Data pane and navigate to

Column tools → **Properties** → **Data category**

Options for data category include City, County, Country, Postal code. This might be worthwhile doing if the location data is not displaying on a map visualisation as expected.

Filled maps

Another map option is a **Filled map**. This visual allows you to fill regions of the map (e.g. countries, counties, states) based on a (location) field.

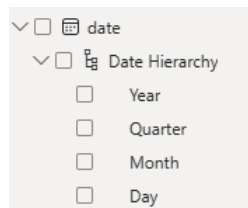
To colour the fillings based on a field (e.g. sum of sales ≥ 1400 in green, and sum of sales ≤ 1400 in red) use the conditional formatting option ($f(x)$) within the "Colours" section of the Format pane.

Sum of amount by location



Creating hierarchies

When importing a field containing dates, a date hierarchy is usually created automatically (i.e. year, quarter, month, day).



You might also want to create custom hierarchies (e.g. country, county, town, postal code).

To do this, right-click on a field within the Data pane (the field you want to be the top level of the hierarchy) and 'Create hierarchy', then right-click on other fields you wish to add to the hierarchy and 'Add to hierarchy'.

To change the order of elements within a hierarchy, go to the "Model View", select the hierarchy from the Data pane, and re-order the fields from within the Properties pane.

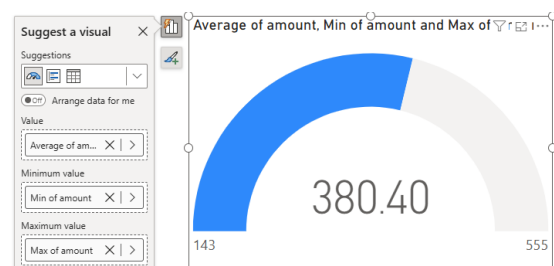
1.7 Measuring performance (KPIs, gauges, and cards)

Performance measuring tools such as KPIs, gauges and cards can be added to reports. They can be added via

Insert → **Visuals** → **Gauge, Card, and KPI**

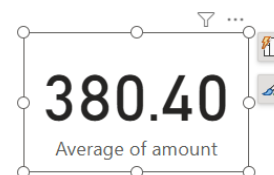
Gauges

A gauge chart uses a circular arc to measure a value relative to a minimum, maximum and target.



Cards

Cards allow you to show a single value, such as a date, location, or numerical value. These differ from text boxes as they interact with slicers/filters etc.

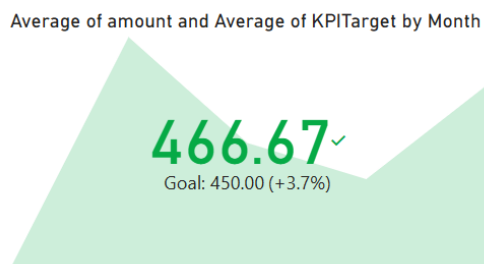


Conditional format table columns

You can add conditional formatting to columns in a table visual. This is done by right-clicking on the field in the "Columns" section of the Build pane, and going to conditional formatting. You have options to add background or font colour based on rules/gradient, and more.

KPI

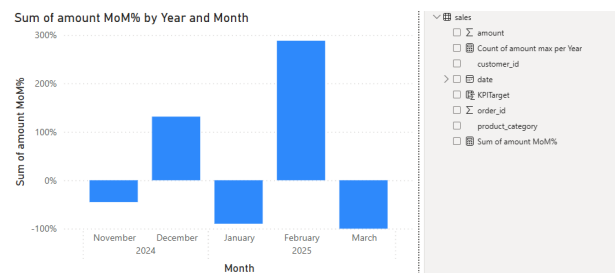
KPI (key performance indicator) displays the progress of a value in a field against a goal over time. Add the value you want to track the progress of to the “Value” section, the time-related field (e.g. Date) to the “Trend Axis” section and the target value to the “Target” section of the Build pane.



1.8 Identify patterns & trends

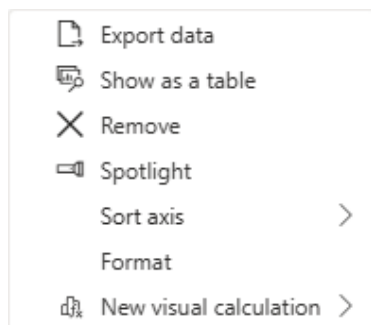
Quick measures

Within the build pane of a visual, click the right arrow next to a field and click New measure. Select a measure (e.g. “Year-to-date total”, “Month-over-month % change”). This will modify the chart to represent the measure.



Exporting data

Click on “More options” (⋮) on a visual to export as a .csv (or similar).



Reference lines and Error bars

Reference lines can be added to a chart from within the Format pane. Navigate to “Reference line”, add a new line, and set its value, colour etc.

There is also an option to add “Error bars”. The error bars can be set to an absolute value or a percentage above and below. There is also the option to add error labels.

Smart narratives/visual summaries

Right-clicking a visualisation and clicking ‘Summarise’ will auto generate a summary of the visualisation.

It is dynamic, meaning it will change based on filters and slicers etc. Note that extra text can be manually added.

Grouping (binning) values in a field

Right-clicking a field in the Data pane gives the option to create a ‘New Group’.

This can be done for categorical, as well as continuous/numerical data. In the case of categorical data, the term ‘group’ is used, whereas for continuous data the term ‘bins’ is used.

Right-clicking the new field (i.e. the grouped/binning data) gives the option to ‘Edit groups’, where the field name can be changed, bin size can be changed, and more.

Groups

Name * amount (bins)

Field amount

Group type Bin

Bin type Size of bins

Min value 143

Max value 555

Binning splits numeric or date/time data into equally sized groups. Enter bin size.

Bin size * 100

Reset to default

OK Cancel

Analyse feature

Right clicking on a data point within a chart gives the option to ‘Analyse’. This is an AI tool that attempts to explain the datapoint, e.g. “Explain the increase”.

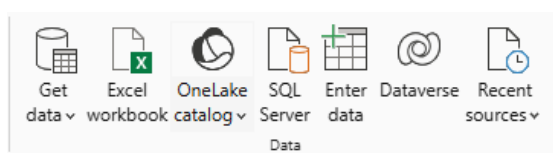
Chapter 2

Getting Data (PowerQuery)

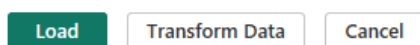
The focus of this chapter is to see how to clean and transform imported data before use in building reports.

Data can be loaded from various sources (e.g. Excel workbooks, databases, CSVs). This is done via:

Home → Data → Get data → ...



Upon selecting a file, you can choose which tables from the file you wish to load. Furthermore, you can choose whether to load directly (this will load it into the Data pane) or 'Transform Data'. Choosing 'Transform Data' will open the data in PowerQuery.



2.1 Introduction and Home tab

All formulas discussed in this section can be found in the Home tab. Thus, all paths to formulas in this section will omit **Home**.

M-language

Formulas in PowerQuery use the M language. Documentation on this can be found at

<https://learn.microsoft.com/en-us/powerquery-m/>

However, for most beginner to low-intermediate data manipulation, the necessary functions can be found in the ribbon somewhere.

Managing columns and rows

To permute the order of the columns, simply drag and drop the column headers.

To manually select a subset of the columns to keep, navigate to **Manage Columns** → **Choose Columns...** → **Choose Columns**

To either remove all selected columns, or remove all non-selected columns, navigate to **Manage Columns** → **Remove Columns...**

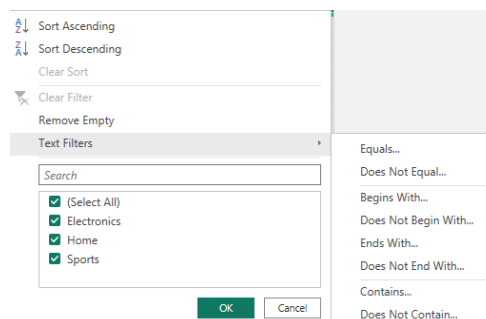
For rows, there are options to keep a subset of the rows based on criteria such as removing duplicates, keeping top N, keeping alternate rows, and more. These options can be found at **Reduce Rows**.

To set the first row as a header (or set the header as the first row), navigate to **Transform** → **Use First Row as Headers...**

Sorting and Filtering

You can sort by a column (ascending/descending) via the drop-down in the column header. The option to remove sorting can also be found here.

Options to filter are also found in the drop-down of a column header. This can be done using data filters or by manually filtering.



Splitting columns

It is sometimes desirable to split a column using a delimiter (e.g. splitting up a file path using “\”, so each directory, and the file itself, get their own column). To do this, navigate to

Transform → Split Column... → By delimiter

There are also options to split by number of characters, digit to non-digit and more.

split on '/'

date	date.1	date.2	date.3
09/10/2024	09	10	2024
09/10/2024	09	10	2024
12/10/2024	12	10	2024
09/11/2024	09	11	2024
09/11/2024	09	11	2024
12/12/2024	12	12	2024
12/12/2024	12	12	2024
12/12/2024	12	12	2024
09/01/2025	09	01	2025
28/02/2025	28	02	2025

Group by

To group the data by the values in a column, select the column and navigate to

Transform → Group by

You can then select aggregate functions to apply to the other columns, for example, count, sum, and/or maximum.

Group By

Specify the column to group by and the desired output.

☒ Basic ☐ Advanced

customer_id

New column name: TotalSpend Operation: Sum Column: amount

2.2 Getting multiple files

Merging tables (analogous to SQL join)

To join two tables along a column, import both tables as separate queries. Then navigate to

Combine → Merge Queries

Choose the queries you wish to merge, as well as the merge column and the join type (inner, outer, left, right etc).

Appending queries (vertically stacking)

To append two or more queries navigate to

Combine → Append Queries

A great use case for this is appending multiple monthly data tables into one large yearly data table.

Importing all data from a folder

To import multiple files from a particular folder, navigate to

New Query → New Source... → More... → Folder

Open the desired folder in PowerQuery via “Transform data”. Then click into ‘Content’ column (containing Binary data) and navigate to **Combine → Combine files**.

2.3 Transform tab

In this subsection, all paths to buttons will omit the tab **Transform**.

Setting the data type of a column

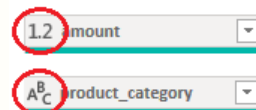
You can set the data type of a given column via

Any Column → Data type

There are also options within **Any Column** to ‘Rename’ the columns, ‘Replace Values/Errors’, and ‘Move’ columns.

Note that almost all of this can be done within the table itself.

For example, to change the data type you can click on the data icon in the column header.



Filling a column

To fill a column in, either upwards or downwards, navigate to

Any Column → Fill

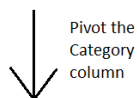
A common use case is filling down a date column where, initially, only one row per day is labeled with a date, while the rows below are blank (implicitly belonging to the same day).

Pivoting a column

Pivoting transforms the data by converting distinct values from a selected column into column headers. Another column is chosen as the value column, along with an aggregate function. The rows and columns of the resulting pivot table function act as a coordinate system, where the aggregated values from the value column are displayed at the intersection of each row and column.

Any Column → Pivot Column

ABC 123	Customer ID	ABC 123	Category	ABC 123	Amount
1	C001		Sports		454
2	C003		Electronics		388
3	C003		Electronics		259
4	C002		Home		283
5	C001		Sports		322
6	C001		Electronics		549
7	C002		Home		401
8	C003		Sports		450
9	C001		Electronics		143
10	C002		Home		555



ABC 123	Customer ID	ABC 123	Sports	ABC 123	Electronics	ABC 123	Home
1	C001			776		692	null
2	C002			null		null	1239
3	C003			450		647	null

To get the intended output, columns that are not involved should be removed first.

Pivoting in PowerQuery and loading into PowerBI is (roughly) equivalent to loading the data directly in PowerBI and using a matrix visualisation.

If the ultimate aim is to display the pivot table, a matrix visualisation is usually the better choice as it will interact better with the other visuals.

Unpivoting columns

The partial inverse operation to pivoting. This converts the column headers of all selected columns into values within a single column and places the corresponding values in a separate column.

Any Column → Unpivot Column

In general, pivoting followed by unpivoting does not always restore the original data. This happens when multiple rows in the original data are identical. Once pivoted, it is impossible to separate them when unpivoting.

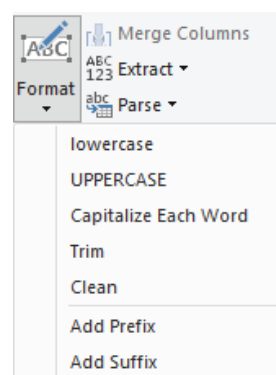
2.4 Text and Numbers

The **Text Column** and **Number Column** sections of the **Transform** tab are only accessible if the selected column's datatype is compatible.

Text Column - Split and Format

The **Split Column** function allows you to split a column into multiple columns via delimiter, number of characters, lower-to-uppercase and more.

There are also some simple format options, such as changing all values in the column to lower case, UPPER CASE, CamelCase.



Text Column - Merge columns

You can merge multiple selected columns using a separator (such as a space, comma, word, or custom).

This removes the existing columns and replaces them with the new column.

If you want to keep the old columns as well, use **Add Column** → **From Text** → **Merge Columns** instead.

Text Column - Extract

You can extract text-based information from a selected column. For example, extract only the text before a given delimiter, between two delimiters, length of the text.

Text Column - Parse

You can parse XML and JSON data. This will unpack the data into a table format that can be worked with within PowerQuery.

Number Column

There are no surprises in the **Number Column** section.

There are standard **Statistics** such as mean, median, standard deviation, etc.

There is a **Standard** option, allowing you to add, subtract, divide, integer divide, etc.

The **Scientific** option provides operations such as power, square root, factorial etc.

There are also **Trigonometry** and **Rounding** functions available

2.5 Dates and Time

Creating lists of dates

Dates are inputted to the formula bar as `#date(yyyy,mm,dd)`.

To create a list of dates, insert the following into the formula bar

```
= List.Dates(#date(yyyy,mm,dd), num_dates:int,
             #duration(days:int,hours,mins,secs))
```

where the `#duration` argument is the increment.

```
= List.Dates(#date(2000,01,01),
              5,
              #duration(1,0,0,0)
              )
```

	List
1	01/01/2000
2	02/01/2000
3	03/01/2000
4	04/01/2000
5	05/01/2000

Similar formulas exist for creating a list of times (`List.Times`) and datetimes (`List.DateTime`).

Transforming dates

Various options for transforming dates and times (in place) can be found at

Transform → Date & Time Column

There are options to extract the day of the week, month, quarter, year, days from [another date] etc.

If you want to keep the old columns as well, you can find the same functions from within the **Add Column** tab.

Dealing with different date formats

To ensure that dates are parsed in the desired format, e.g. UK dates dd/mm/yyyy, US dates mm/dd/yyyy, or otherwise, click on the date column header's datatype icon and navigate to "Using Locale...". This allows you to convert to date using a specified locale/region.

Change Type with Locale

Change the data type and select the locale of origin.

Data Type

Date

Locale

English (United Kingdom)

2.6 Adding Columns

Adding a column from examples

The computer can help work out a formula based on some example inputs and outputs. This will also supply the function needed to produce the output.

General → Column From Examples

For example, if you wanted to produce a column function that multiplied by two, you could supply the input column 1,2,3,4 and the output column 2,4,6,8.

Adding conditional columns

You can add new columns based on conditions imposed on existing ones. This is found at:

General → Conditional Column

For example, you could use this function to easily add a column saying "OK" if a value in an existing column is greater than 0, otherwise "Not OK".

Index columns

You can add an index column by navigating to

General → Index Column

The ‘Custom’ option allows you to choose your own starting index and increment value.

2.7 Converting text to number (dealing with locale)

Different locales might have different number formatting, e.g. ‘.’ vs ‘,’ as the decimal separator.

To deal with this cleanly, you can use the **Number.FromText** function, which accepts a locale as an optional argument (e.g. en-GB, en-US, fr-FR, es-ES).

Custom Column

Add a column that is computed from the other columns.

New column name

UKNumber

Custom column formula ⓘ

= Number.FromText([NumberAsText], "en-GB")

2.8 View and Help tabs

The applied transformation steps, as well as the formula bar, can be enabled or disabled via a checkbox by navigating to

View → Layout

Within the **Data Preview** section, the **Column quality** checkbox will show the percentage of errors in each column. This is good for checking if the data is of good quality. Another useful checkbox is **Column profile**, which gives stats on a selected column, such as the number of unique values, the min and max, and the number of NaNs.

Query Dependencies, found at

View → Dependencies → Query Dependencies

will open a window showing how all queries depend on each other.

Parameters and functions

Logic that is used multiple times (especially if over multiple queries) might be best suited to being defined as a function.

An easy way to create a new function is to make a blank query (**Home → New Query → Enter Data**), then navigate to **View → Advanced → Advanced Editor**, where the function can be written, it is of the form

```
let
    MyFunction = (arg1, arg2, ...) =>
        //function logic
in
    MyFunction
```

Pressing “Done” will convert the blank query into a function (with icon “fx”). For example, the function to add two arguments is

```
let
    AddTwo = (num1, num2) =>
        num1 + num2
in
    AddTwo
```

2.9 Importing from SQL Server SQL Server

When importing from an SQL Server (**Home → New Query → New Source... → SQL Server**), there are two options for ‘Data Connectivity mode’, namely ‘Import’ and ‘DirectQuery’.

The ‘Import’ option loads a copy of the database, whereas ‘DirectQuery’ is a live link to the database, which queries the data directly from the source when needed (this is good if the source data is updated regularly).

It is also recommended that you check the “Include Relationship Columns” option. This imports each table that is connected to the imported data as a column, which can then be expanded in PowerQuery.

If data is loaded into PowerQuery, right clicking a step in the “Query Settings” pane allows you to view the Native (SQL) Query. This will show the SQL query used to obtain the displayed data.

Chapter 3

Modelling Data

3.1 Introduction to models

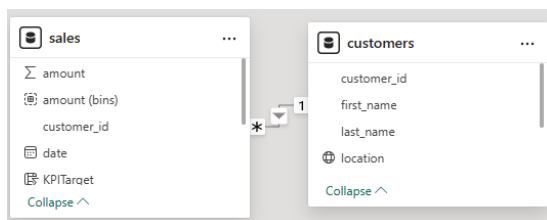
In summary, modelling the data involves creating relationships between different tables, defining new calculated measures using DAX for efficient calculations, setting up data hierarchies (e.g. Year, Month, Day).

Creating relationships

If you have multiple data sources loaded, you can create links between them by going to Model View, and then clicking and dragging a field within a table onto another field within a different table.

The links will be labelled with (1, *), (*, 1), or (*, *), which denotes the relationship type. For example, (1, *) denotes a one-to-many relationship. The arrow signifies the direction the data “flows” in.

Double-clicking on an arrow allows you to edit.



Here, each customer is involved in many sales, therefore, we have a one-to-many relationship flowing from the customers table to the sales table.

3.2 Logical DAX functions

Calculated columns and measures are created using the DAX language. There are around 230 DAX functions, and info on them can be found at:

<https://learn.microsoft.com/en-us/dax/dax-function-reference>

Adding a calculated column

Within the Model view, new columns can be added to a table by selecting that table (via clicking on it, or selecting it from the Data pane) and navigating to

Home → Calculations → New column

Alternatively, right-clicking on a table supplies the option to create a new column.

For example, to create a column in the Sales table showing Amounts in USD, you would right-click the ‘Sales’ table, select ‘New column’ and type the following into the formula bar:

```
amountUSD = Sales[amount] * 1.33
```

When typing formulas, autocomplete suggestions will appear (such as formula names, existing table names, and existing column names within those tables). These are chosen and selected using the ‘Arrow keys’ and ‘Tab’.

IF and BLANK

The IF function has syntax:

```
IF(<logical_test>, <value_if_true>[,  
    <value_if_false>])
```

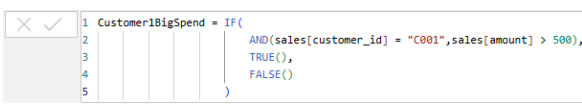
where the logical test usually involves one or more columns from the data.

The analog of null/None/NaN in DAX is BLANK(). Moreover, ISBLANK() checks if a cell is blank or not. Blank cells are excluded from counts, average calculations, etc.

AND, OR, and NOT

- **AND(<logical1>,<logical2>)**
returns TRUE if both logical statements are TRUE, otherwise returns FALSE.
- **OR(<logical1>,<logical2>)**
returns TRUE if at least one of the logical statements is TRUE, otherwise returns FALSE.
- **NOT(<logical>)**
changes TRUE to FALSE, and FALSE to TRUE.

For example, to create a column that shows TRUE if Customer 1 (with ID C001) made a purchase of over 500, and FALSE otherwise:



Formula: `Customer1BigSpend = IF(AND(sales[customer_id] = "C001", sales[amount] > 500), TRUE(), FALSE())`

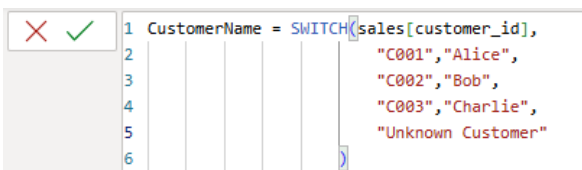
order_id	date	customer_id	amount	Customer1BigSpend
1	03 October 2024	C001	454	False
2	03 October 2024	C003	388	False
3	22 October 2024	C003	259	False
4	03 November 2024	C002	283	False
5	05 November 2024	C001	322	False
6	12 December 2024	C001	549	True
7	12 December 2024	C002	401	False
8	12 December 2024	C003	450	False
9	09 January 2025	C001	143	False
10	28 February 2025	C002	555	False

SWITCH

This is essentially an IF, ELSE IF, ..., ELSE function, all wrapped up into one. The syntax is as follows:

SWITCH(<expression>, <value>, <result>[, <value>, <result>]...[, <else>])

For example, if you wanted to manually add a column with the customers' names to the Sales table:



Formula: `CustomerName = SWITCH(sales[customer_id], "C001", "Alice", "C002", "Bob", "C003", "Charlie", "Unknown Customer")`

order_id	date	customer_id	CustomerName
1	03 October 2024	C001	Alice
2	03 October 2024	C003	Charlie
3	22 October 2024	C003	Charlie
4	03 November 2024	C002	Bob
5	05 November 2024	C001	Alice
6	12 December 2024	C001	Alice
7	12 December 2024	C002	Bob
8	12 December 2024	C003	Charlie
9	09 January 2025	C001	Alice
10	28 February 2025	C002	Bob

3.3 Stats and measures

Creating new measures

New measures can be created via navigating to

Home → Calculations → New measure

However, an alternative (and probably better, in terms of ensuring they are stored in the correct place) way to create measures is to right-click the desired table in the Data pane, and select "New measure".

Some standard functions for measures are SUM, AVERAGE, COUNT, COUNTBLANK, MIN, MAX.

Extended functions (X functions)

For most of the standard functions above, there exists an "extended version", given by appending an "X" to the function name, e.g. SUMX, AVERAGEX, MINX, MAXX, ...

These extended functions accept a table and an expression, and apply the stated function to the value of each row after applying the expression.

As a psuedo example, $\text{MAXX}([1, 2, -3], x \mapsto x^2)$ would return the value 9, as it is finding $\text{MAX}(1^2, 2^2, (-3)^2) = \text{MAX}(1, 4, 9)$.

Use of extended functions is preferred to creating "helper columns".

Statistical Functions

There are many functions related to distributions, such as Chi-squared (CHISQ.DIST), and normal (NORM.DIST), among others.

You can find percentiles of numeric data (using PERCENTILE.INC, PERCENTILE.EXC, as well as the extended versions).

You can rank the rows using RANK, RANKX, or RANK.EQ.

Microsoft Support states that RANK is still available for compatibility with older versions of Excel, but recommends using RANK.EQ for current work.

3.4 Mathematical Functions

Some useful math-based functions are

- **CEILING**(<number>, <significance>)
rounds a number up to the next multiple of significance.
- **FLOOR**(<number>, <significance>)
rounds a number down to the next multiple of significance.
- **INT**(<number>)
rounds a number down to the nearest integer
- **ROUND**(<number>, <num_digits>)
rounds a number to the specified number of digits.

A positive value for <num_digits> rounds to the right of the decimal place, whilst a negative value rounds to the left of the decimal place.

For example, **ROUND**(111.11,-2) rounds to the nearest 10^2 , and so returns 100.

- **MROUND**(<number>, <multiple>)
rounds a number to the nearest multiple of <multiple>.

For example, **MROUND**(0.77, 0.3) = 0.9.

- **MOD**(<number>, <divisor>)
returns the remainder when <number> is divided by <divisor>
- **QUOTIENT**(<numerator>, <denominator>)
returns the integer portion of the fraction $\frac{\text{<numerator>}}{\text{<denominator>}}$.
- **SIGN**(<number>)
returns the sign of the number
(-1 if negative, 1 if positive, 0 if zero).
- **ABS**(<number>)
returns the absolute value of a number.
- **EXP**(<number>)
returns $e^{\text{<number>}}$
- **SQRT**(<number>)
returns the square root of <number>
- **POWER**(<number>, <power>)
returns $\text{<number>}^{\text{<power>}}$

3.5 Text Functions

Some useful text-based functions are

- **FIND**(<find_text>, <text>[, [<start_num>] [, <NotFoundValue>]])
returns the starting position of a text string within another text string.

There is the option to start from a specific <start_num> and return a custom <NotFoundValue> if the given text string isn't found.

FIND is case-sensitive.

- **SEARCH**(<find_text>, <text>[, [<start_num>] [, <NotFoundValue>]])
almost the same as FIND, except SEARCH is case-insensitive.
- **SUBSTITUTE**(<text>, <oldTxt>, <newTxt>, <instance_num>)
Replaces existing text with new text in a text string.

The optional argument allows you to specify which occurrence of the existing text you want to replace. If omitted, every instance of the existing text is replaced

- **LEFT**(<text>, <num_chars>)
extracts the specified number of characters from the start of the text string
- **LEN**(<text>)
returns the number of characters in the text string
- **REPLACE**(<text>, <start_n>, <n_chars>, <new_text>)
replaces part of a text string, based on the number of characters specified, with a different text string.
- **CONCATENATE**(<text1>, <text2>)
Joins two text strings into one text string.
- **LOWER**(<text>)
converts all letters in a text string to lowercase.
- **TRIM**(<text>)
removes all spaces from text except for single spaces between words.

Datatype conversion functions

The following functions get values into a specified format and return the result as a different datatype.

- **FIXED**(<number>, <digits>, <no_commas>)
rounds a number to the specified number of decimals and returns the result as text.
If the optional argument <no_commas> is set to 1, then the resulting string WILL NOT have comma separators.
If the optional argument <no_commas> is set to 0, then the resulting string WILL have comma separators
- **VALUE**(<text>)
converts a string representing a number to the decimal data type.
- **FORMAT**(<value>, <f_string>[, <locale>])
converts a value to text according to the specified format string. For example,

EVALUATE
VAR NumberValue = 12345.67
RETURN
UNION(
ROM("Format Type", "General Number", "Formatted Value", FORMAT(NumberValue, "General Number")),
ROM("Format Type", "Currency", "Formatted Value", FORMAT(NumberValue, "Currency")),
ROM("Format Type", "Fixed", "Formatted Value", FORMAT(NumberValue, "Fixed")),
ROM("Format Type", "Standard", "Formatted Value", FORMAT(NumberValue, "Standard")),
ROM("Format Type", "Percent", "Formatted Value", FORMAT(NumberValue, "Percent")),
ROM("Format Type", "Scientific", "Formatted Value", FORMAT(NumberValue, "Scientific"))
)

Results Result 1 of 1 Copy

	[Format Type]	[Formatted Value]
1	General Number	12345.67
2	Currency	\$12,345.67
3	Fixed	12345.67
4	Standard	12,345.67
5	Percent	1234567.00%
6	Scientific	1.23E+04

The options for 'format_string' are vast, see more details in the docs: <https://learn.microsoft.com/en-us/dax/format-function-dax#predefined-numeric-formats>

3.6 Look Up Value function

- **LOOKUPVALUE** (<result_columnName>, <search_columnName>, <search_value> [, <search2_columnName>, <search2_value>]... [, <alternateResult>])
returns the entry in the result column corresponding to the search value(s) from the search column(s). This is somewhat analogous to XLOOKUP from Excel.

3.7 Filter and Value functions

- **RELATED**(<column>)
returns a related value in the specified column from another table.

The RELATED function requires that a relationship exists between the tables.

You specify the column that contains the data that you want, and the function follows an existing many-to-one relationship ("upstream") to fetch the value from the specified column in the related table.

For example, to add the customers' first names to the dummy Sales table (note the existing relationship between customer_id columns), you can do

1 CustomerName = RELATED(customers[first_name])

order_id	date	customer_id	amount	CustomerName
1	03 October 2024	C001	454	Alice
2	03 October 2024	C003	388	Charlie
3	22 October 2024	C003	259	Charlie
4	03 November 2024	C002	283	Bob
5	05 November 2024	C001	322	Alice
6	12 December 2024	C001	549	Alice
7	12 December 2024	C002	401	Bob
8	12 December 2024	C003	450	Charlie
9	09 January 2025	C001	143	Alice
10	28 February 2025	C002	555	Bob

- **RELATEDTABLE**(<tableName>)
returns a filtered table using the one-to-many relationship. This is used to "travel downstream" along the relationship. This is usually combined with an aggregation function on the resulting filtered table.

For example, to obtain a column in the Customers table showing the total spend for each person, you can do

1 TotalSpend = SUMX(RELATEDTABLE(sales), sales[amount])

customer_id	first_name	last_name	location	TotalSpend
C001	Alice	Anderson	London	1468
C002	Bob	Bennett	Manchester	1239
C003	Charlie	Clarke	Newcastle	1097

RELATED travels "upstream" whereas RELATEDTABLE travels "downstream" in relation to the direction of the relationship.

- **ALL**(<table> | <column>)
returns all the rows in a table, or all the values in a column, ignoring any filters that might have been applied.
- **FILTER**(<table>, <filter>)
returns a filtered table. The argument <filter> is a Boolean expression, e.g. `Sales[amount] > 0`.
- **CALCULATE**(<expr>[, <filter1> [, <filter2> [, ...]]])
calculate an expression in the specified (filtered) context.

For example, to create a measure of Alice's total spend, you can do

```
AliceSales = CALCULATE(
    sum(sales[amount]),
    customers[first_name]="Alice"
)
```

1468

AliceSales

- **ALLEXCEPT**(<table>, <column> [, <column> [, ...]])
removes all filters in the table except filters that have been applied to the specified columns.

3.8 Date Intelligence functions

Some date functions are

- **DATE**(<year>, <month>, <day>)
returns the specified date in datetime format
- **DATEVALUE**(date_text)
converts a date in text format to a date in datetime format
- **YEAR**, **MONTH**, **DAY**, **SECOND**, **MINUTE**, ...
all accept a <date> argument. They extract the relevant piece of information from the date.

The equivalent of MIN and MAX for dates is

- **FIRSTDATE**(<dates>)
returns the first date from a range of dates
- **LASTDATE**(<dates>)
returns the last date from a range of dates

To obtain dates for a given period you can use

- **DATESINPERIOD**(<dates>, <start_date>, <number_of_intervals>, <interval>)
Returns a table that contains a column of dates from a <dates> column in the period beginning from the specified start date and continuing for the specified number and length of intervals.
- **DATESMTD**(<dates>)
Returns a table containing a column of the month to date, in the current context.
- **DATESQTD** (quarter to date), and **DATESYTD** (year to date) work similarly.

3.9 Calculated Tables & Visuals

New calculated tables

From the Model View, navigate to

Modelling → Calculations → New Table

to create a new table (using a DAX expression).

For example, you can merge, join, group by and take unions of existing tables here (if you'd rather do it here than in PowerQuery).

Some relevant functions are

- **UNION**(<table_expression1>, <table_expr2> [, <table_expr>]...)
returns a union of the tables.
- **NATURALINNERJOIN**(<LeftTable>, <RightTable>)
performs an inner join of the two tables.
- **EXCEPT**(<table_expr1>, <table_expr2>)
rows in the first table that are not in the second.

Adding Calculations to Visuals

Within the Report View, select a visualisation and navigate to

Home → Calculations → New Visual calculation

to add a new calculation to the visual. Some common examples are 'Running Sum', 'Moving Average', 'Percent of Grand Total', 'Versus Next'.

This calculation is usable only within this visualisation (i.e. it's not visible in the Data pane).

Moreover, only data currently in the visualisation can be used in calculations (i.e. external data from the Data pane cannot be used).

Chapter 4

Sharing Data (The Power BI Service)

Key terminology

- Data source - the source of the data... this could be local (e.g. a .csv file) or cloud-based.
- Semantic model - a model capturing the relationships between data.
- Report - a set of visualisations arising from a model.
- Dashboard - built from “tiles” i.e. visualisations from (multiple) reports. Dashboards can be created by pinning visualisations from within a report.
- Workspace - a collection of models, reports and dashboards.
- Apps - a published collection of related reports and dashboards.

Uploading data

Within a workspace, click ‘New’ → ‘Semantic model’. Excel files (.xlsx), and CSVs can be uploaded this way. Excel data needs to be formatted as a table.

4.1 Row level security

When sharing reports with others, you might wish to limit the rows of the data that they can see (e.g. different departments should only be able to see data relevant to their department).

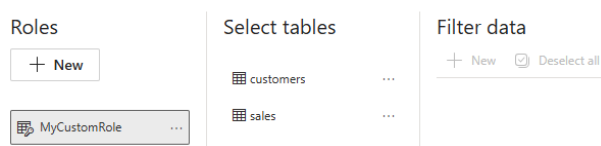
Row-level security is created in the desktop app. To add it, navigate to

Modelling → Security → Manage roles

Then create a new role, and for each table, add DAX expressions to filter the data, leaving only the data that users assigned to the role should be able to see.

Manage security roles

Create new security roles and use filters to define row-level data restrictions.



Assigning users to the roles is done within the Power BI Service. Navigate to the workspace, and click on “...” next to the semantic model, then, click on security to assign users to roles.

Dynamic roles can be created, which filter based on the user’s email address, using the DAX expression `USERPRINCIPALNAME()`.

4.2 Dashboards

Dashboards vs reports

A report is made using a single semantic model (of course, this can be made up of multiple tables). Dashboards can incorporate data from multiple models. Dashboards are made by pinning visualisations from reports onto them, and visualisations from multiple reports can be pinned onto the same dashboard.

Managing tiles

Tiles (on a dashboard) can be resized by clicking in the bottom right corner of the tile and dragging.

Tiles are NOT live-linked to the report. For example, the visualisation type at the time of pinning is the one displayed in the dashboard, even if it has been later changed in the report.

Clicking on Edit → Add tile allows you to add web content and media, such as a YouTube video.

4.3 Managing Semantic models Apps

Opening a semantic model and navigating to **Export** → **Analyse** will allow you to work with the data in Excel.

Models can be endorsed by going into the model settings, navigating to "Endorsement and discovery" and selecting one of None, Promoted, or Certified.

Data gateways

A data gateway is a middle-man that allows the Power BI Service to communicate with a local network (also known as on-premises) in order to get the latest version of data, for example.

From the Power BI Service, click the download icon and choose "Data Gateway".

To configure a scheduled refresh, ensure the credentials are edited from the "Data source credentials" tab in the workspace settings. Then, from the "Refresh" tab, enable a refresh schedule and select the frequency.

4.4 Managing workspaces

A workspace is a collection of models, reports and dashboards. Workspaces can be created and managed from the "Workspaces" tab on the left-hand side of the Power BI Service.

Clicking "..." next to a workspace in the "Workspaces" window allows you to add users to the workspace. Users can be added as admins, members, contributors, or viewers.

You can save a copy of a report to a different workspace than the one it lives in.

Reports made using a model from a workspace can be saved to a different workspace. An admin can disable this feature from within the admin portal.

An app is a way to nicely package (a subset of) the files in a workspace. Note that you cannot create apps using the files in "My Workspace".

They are created by selecting "Create app" in the top bar from within a workspace. In the pop up window, details such as an app name and logo can be set, as well as some other basic settings.

In the next window, you can select which files from the workspace you wish to add to the app. Finally, an audience is chosen (either the entire organisation or specific users/groups).

After creation, the "Create app" button will be replaced with an "Edit app" button where settings can be changed.

Embedding reports into websites

Within a report, navigate to

File → **Embed Report** → **Website or portal**.

Note that this is a secure embedding, so the viewer must be logged in and have permission to view the report.

To publish a report publicly (for everyone to view), navigate to

File → **Embed Report** → **Publish to web**.

Note that in order to do this, the administrator setting "Allow users to create new embed codes" (found within the admin portal) must be enabled.