Data Action Lab

Sysabee DAVHILL Group IDLEWYLD

Enseignement univ.: 50+ cours; **ateliers:** 40+; **projets:** 60+; **expérience combinée:** 35+ années.
Co-entreprise pré-qualifiée à la liste des fournisseurs I.A. du GdC – EN578-180001/A (1ière bande).

Nouveau catalogue de formation approfondie disponible sur la toile au **data-action-lab.com**

Combined experience: 50+ university courses, 100+ corporate workshops, 60+ projects, 35+ years.
Joint venture qualified for GoC A.I. Source List – EN578-180001/A (Band 1).

New advanced training catalogue available at **data-action-lab.com**

Data Action Lab

What We Do

Training and long courses

Workshops and short courses

Knowledgebase curation

Data labs

Data Action Lab

Training

Training Paths

- Data Novice
- Data Engineer
- Data Practitioner
- Data Scientist
- Data Manager
- Data Champion

Training Learning Interests

- Visualization and Dashboards
- Introduction to Data Science
- Advanced Data Science
- Machine Learning Toolbox
- Spotlight on Classification
- Spotlight on Clustering
- Text Analysis
- Special Topics in AI and DS
- Hands on Data Analysis
- Data Strategy and Governance

Data Action Lab

Expertise

- Business intelligence
- Data visualization design
- Data analytics and data science
- Data engineering
- Advanced statistics and machine learning
- Artificial and augmented intelligence
- Process and systems modeling
- Software implementation and integration

Data Action Lab

Vision

- Provide a space for data consumers, producers, practitioners, scientists and champions to make a place for themselves in the digital world.
- Provide paths for education and enrichment for all these groups.
- Keep pace with developments in the digital arena and keep Data Action Lab participants moving and aligned with these relevant developments.
- Provide just-in-time learning opportunities for Data Action Lab members, focusing on their specific challenges and skillsets.

PBI-1: GETTING TO KNOW POWER BI

DATA ACTION LAB – POWER BI SERIES

IDLEWYLD Sysabee DAVHILL

data-action-lab.com

COURSE OVERVIEW

Course #: PBI-1 **Duration:** 1.5 day
Course Title: Getting to know Power BI

Description:

1. Introduction
    - Why are we doing all of this?
2. Getting Things Ready
    - Preparation and importance of clean data
3. Importing and modeling our first data set
    - Importing, refining and linking data sets

COURSE OVERVIEW

Course #: PBI-1 **Duration:** 1.5 day
Course Title: Getting to know Power BI

Description:

4. Creating out first calculations and charts
    - Calculated columns, measures (and measure tables), mathematical operations, filtering & slicing and logical operations
5. Organizing Data
    - Hierarchies, groups (numerical & text), custom sorting
6. Data Wrangling
    - Removing rows, replacing values, splitting columns, trimming and cleaning, appending tables, choosing what data gets loaded

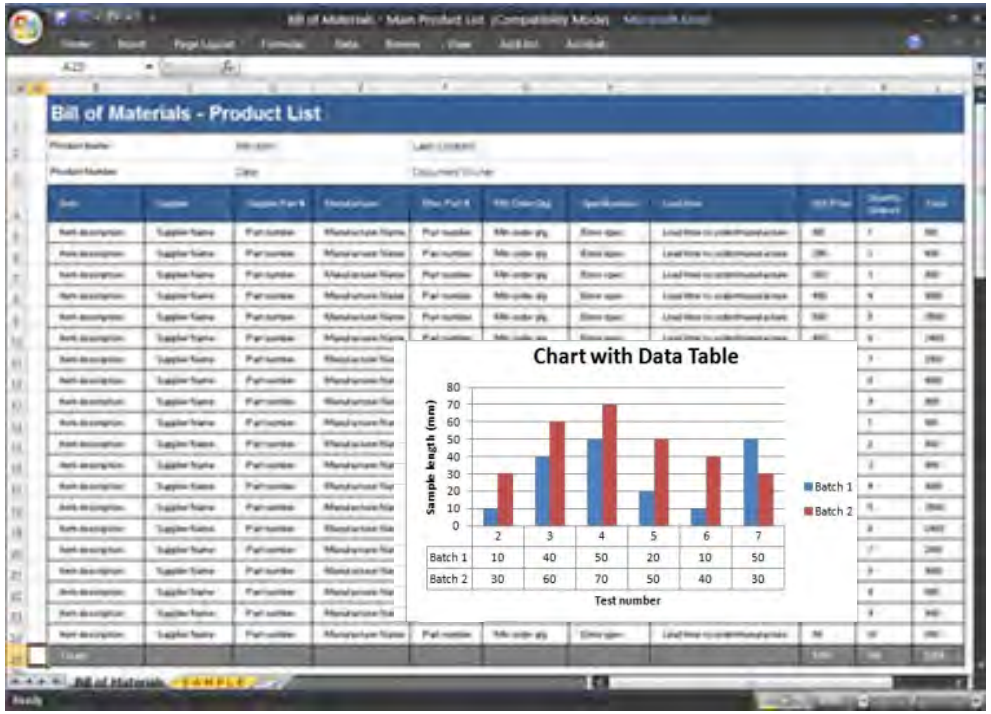
INTRODUCTION

PBI-1: GETTING TO KNOW POWER BI

INTRODUCTION

The past is **data-driven:**

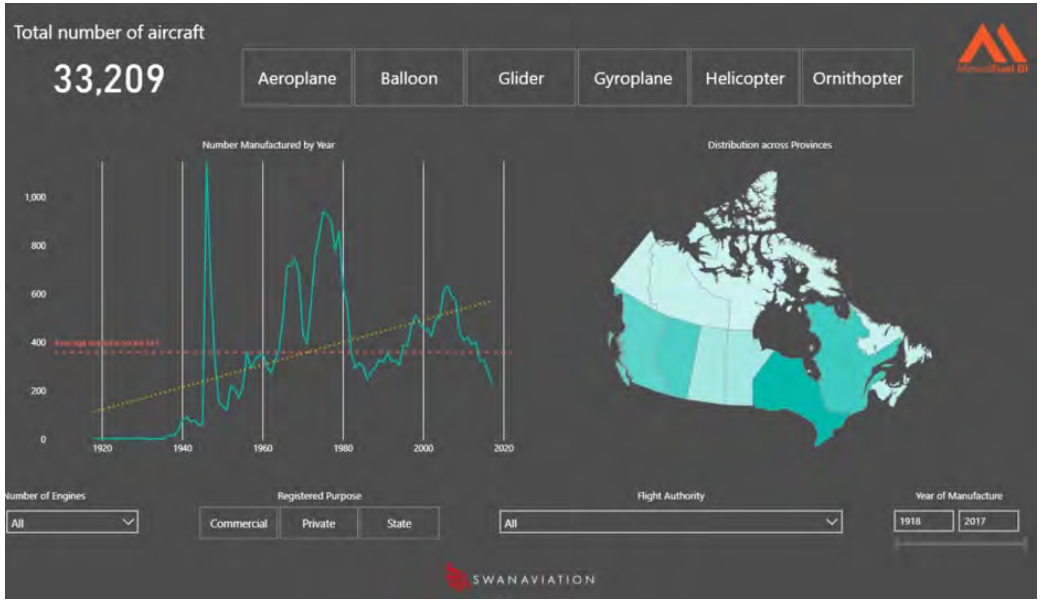
- mostly Excel (or reporting tools like Cognos)
- mostly numbers, tables and non-interactive graphs
- distributed on desktop computers, by email, in PowerPoint presentation
- static, mostly backwards looking (lagging indicators)
- KPIs and dashboards were somewhat contrived



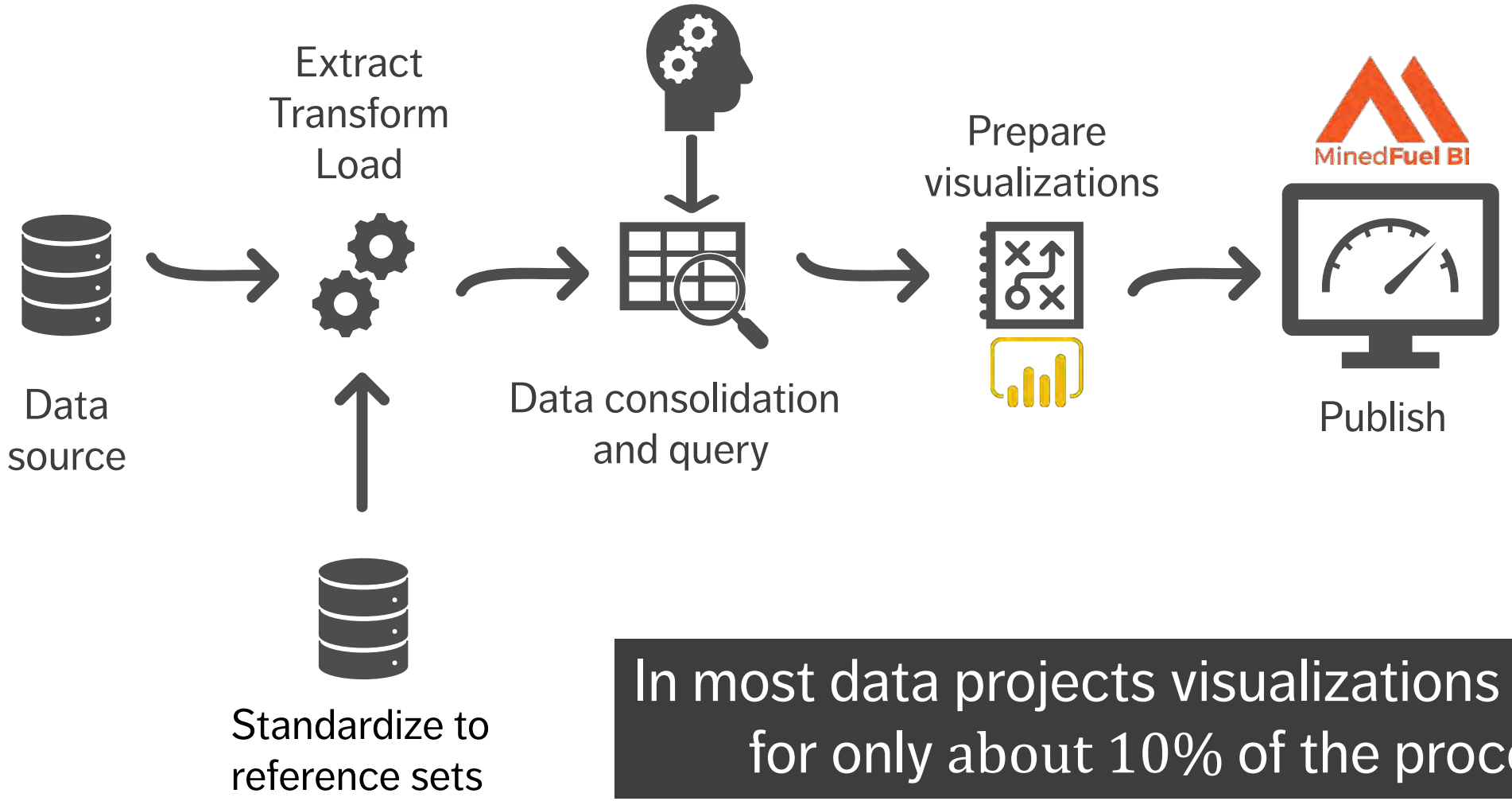
INTRODUCTION

The future is **story-driven:**

- new tools: Power BI, R, Qlickview, etc.
- mostly visualizations, occasional numbers and tables
- distributed on the web (internal and external)
- dynamic and both backwards and forwards looking (leading and lagging indicators)
- data is for everyone



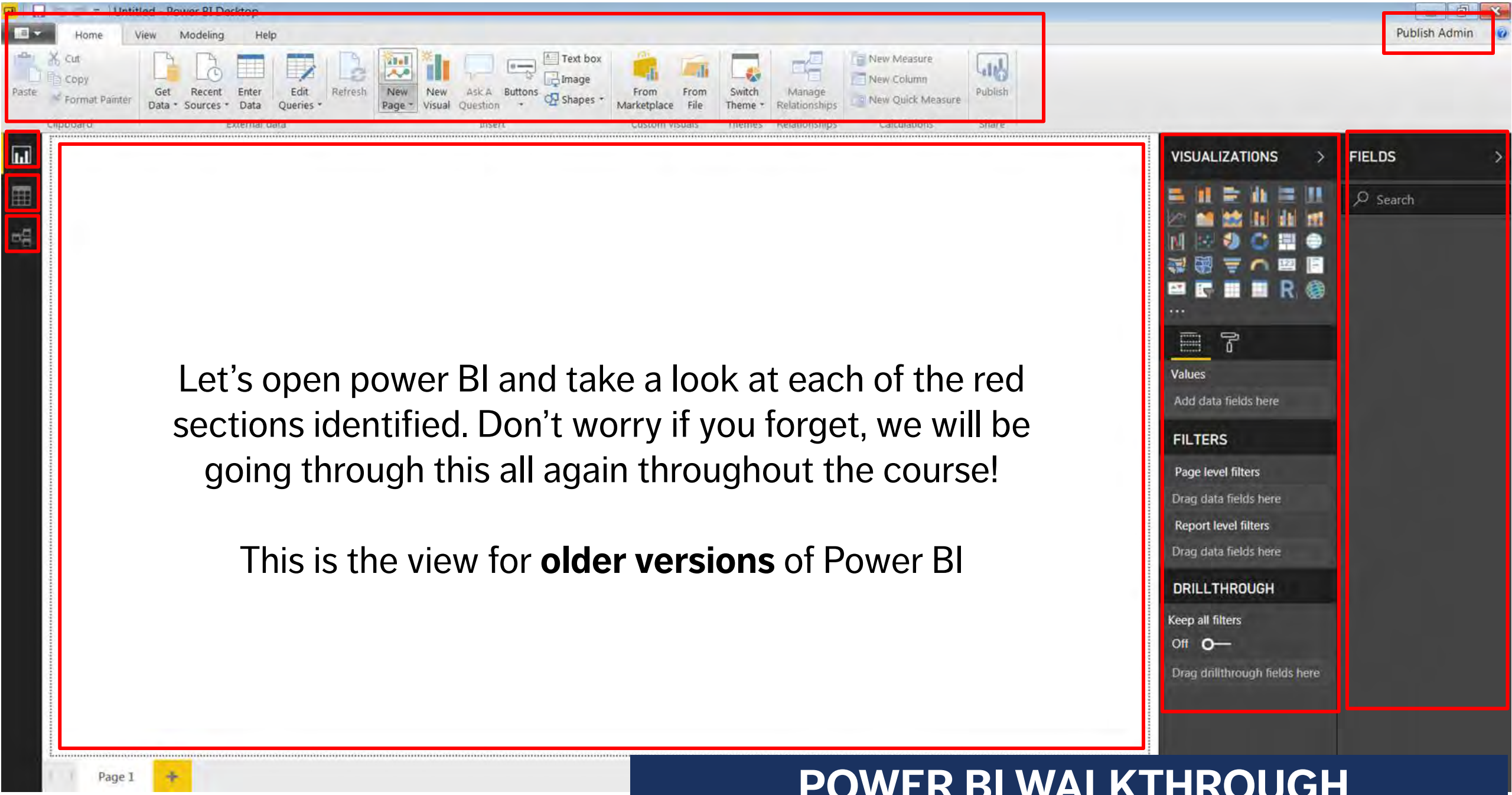
INTRODUCTION



In most data projects visualizations account for only about 10% of the process.

GETTING THINGS READY

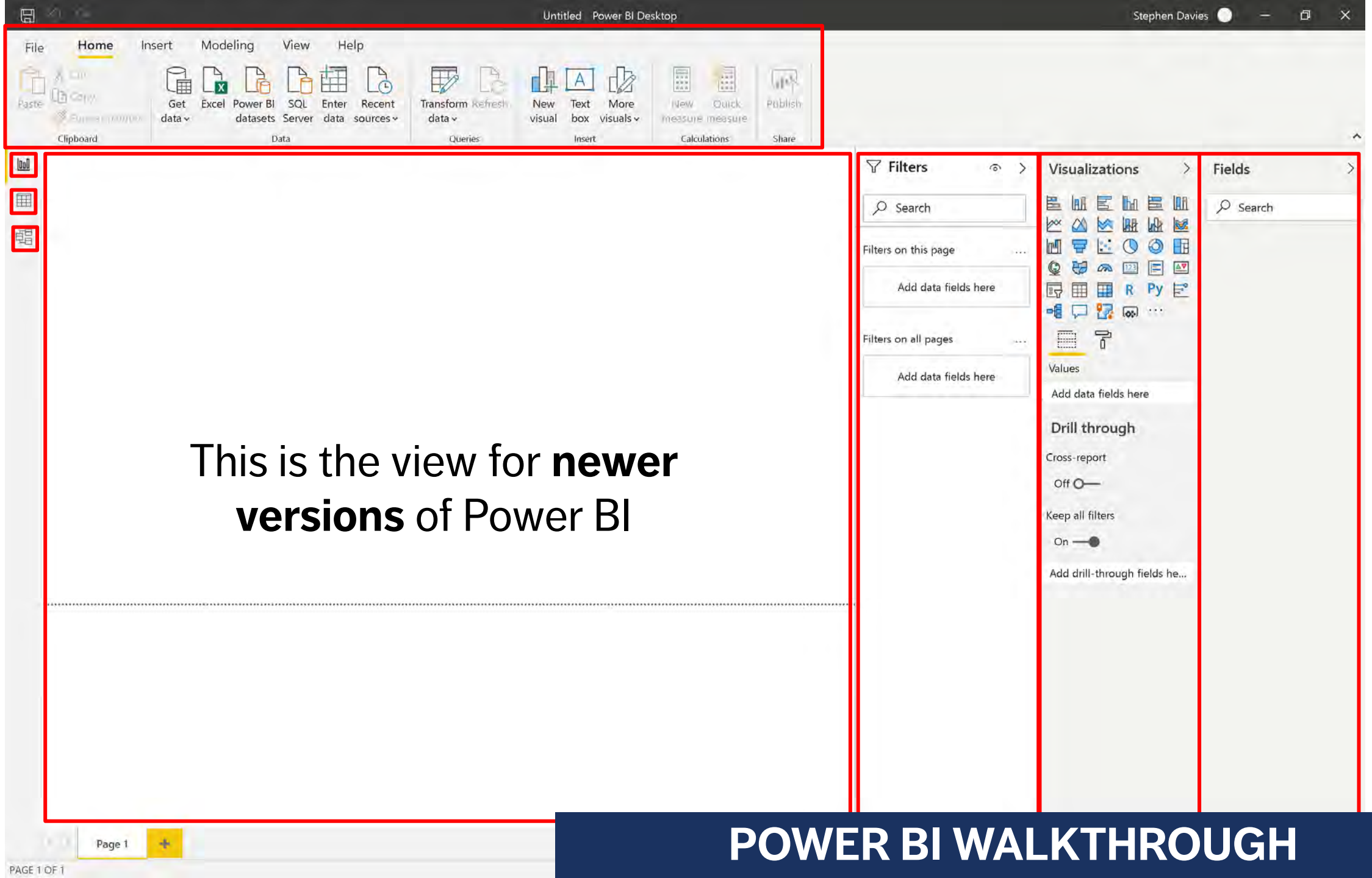
PBI-1: GETTING TO KNOW POWER BI



Let's open power BI and take a look at each of the red sections identified. Don't worry if you forget, we will be going through this all again throughout the course!

This is the view for **older versions** of Power BI

POWER BI WALKTHROUGH


Untitled Power BI Desktop
Stephen Davies
File
Home
Insert
Modeling
View
Help
Paste
Cut
Copy
Format painter
Clipboard
Get data
Excel
Power BI datasets
SQL Server
Enter data
Recent sources
Data
Transform data
Refresh
Queries
New visual
Text box
More visuals
Insert
New measure
Quick measure
Calculations
Publish
Share
This is the view for **newer versions** of Power BI
Filters
Search
Filters on this page
Add data fields here
Filters on all pages
Add data fields here
Visualizations
Values
Add data fields here
Drill through
Cross-report
Off
Keep all filters
On
Add drill-through fields he...
Fields
Search
Page 1
PAGE 1 OF 1
POWER BI WALKTHROUGH

WHY WE CLEAN DATA

BINGO!!!!

random'missing' values	outliers	values'outside'of' expected'range'4 numeric	factors' incorrectly/iconsiste ntly'coded	date/time'values'in' multiple'formats
impossible'numeric' values	leading'or'trailing' white'space	badly'formatted' date/time'values	non4random'missing' values	logical' inconsistencies' across'fields
characters'in' numeric'field	values'outside'of' expected'range'4 date/time	DCB!	inconsistent'or'no' distinction'between' null,'0,not'available,' not' applicable,missing	possible'factors' missing
multiple'symbols' used'for'missing' values	???	fields'incorrectly' separated'in'row	blank'fields	logical'iconsistencies' within'field
entire'blank'rows	character'encoding' issues	duplicate'value'in' unique'field	non4factor'values'in' factor	numeric'values'in' character'field

IMPORTING AND MODELING DATA

IMPORTING AND MODELING DATA

Let's take a look at the data sets that we are going to use for this course (we might use others as well), go to the website URL you have been given and download them to your hard drive:

- "Data set – Accounting.xlsx"

Accounting Control Number	Journal Voucher Type Code	Accounting Effective Date	Journal Voucher Item Amount	Project Identifier
5000091	MC	01-Apr-18	$27,529.28	PR007
5000093	SA	03-Apr-18	$87,508.20	PR009
5000094	O&M	04-Apr-18	$78,590.42	PR010
5000095	MC	05-Apr-18	$157,288.63	PR011
5000096	MIC	06-Apr-18	$61,605.96	PR012
5000097	SA	07-Apr-18	$39,859.12	PR013
5000098	O&M	08-Apr-18	$31,037.60	PR014
5000099	MC	09-Apr-18	$142,056.49	PR015
5000101	SA	11-Apr-18	$44,404.21	PR017
5000102	O&M	12-Apr-18	$72,057.43	PR018

Tab – Accounting Transactions

Code	Description
O&M	Operations and Maintenance
MC	Major Capital
MIC	Minor Capital
SA	Salary

Tab – Journal Voucher Type Code

IMPORTING AND MODELING DATA

- "Data set – Project.xlsx"

Project Code	Project Name	O&M Budget	Salary Budget	Major Cap Budget	Minor Cap Budget	FTE Budget
PR001	Parks	$25,000,000.00	$15,000,000.00	$10,000,000.00	$5,000,000.00	9.00
PR002	Buildings	$50,000,000.00	$8,000,000.00	$100,000,000.00	$50,000,000.00	6.00
PR003	Emergency Response	$3,000,000.00	$7,000,000.00	$8,000,000.00	$3,000,000.00	6.00
PR004	Office	$4,000,000.00	$40,000,000.00	$80,000,000.00	$20,000,000.00	12.00
PR005	Roads	$50,000,000.00	$20,000,000.00	$250,000,000.00	$100,000,000.00	7.00
PR006	Science	$5,000,000.00	$10,000,000.00	$80,000,000.00	$20,000,000.00	7.00
PR007	Heritage	$15,000,000.00	$5,000,000.00	$30,000,000.00	$17,000,000.00	9.00
PR008	Celebration	$2,000,000.00	$4,000,000.00	$0.00	$0.00	7.00
PR009	Research	$5,000,000.00	$12,000,000.00	$25,000,000.00	$5,000,000.00	6.50
PR010	Upgrades	$4,000,000.00	$2,000,000.00	$30,000,000.00	$5,000,000.00	10.00

Tab – Project Tombstone

Project Code	Date	FTE (- out +	Group-Level
PR001	01-Apr-18	2	AS-04
PR002	01-May-18	2	AS-05
PR003	01-Jun-18	2	ENG-01
PR004	01-Jul-18	4	PR-01
PR005	01-Aug-18	3	PA-03
PR006	01-Sep-18	2	AS-02
PR007	01-Oct-18	1	CR-03
PR008	01-Nov-18	3	FI-02
PR009	01-Dec-18	4	FO-03
PR010	01-Jan-19	2	CR-01
PR011	01-Feb-19	3	AS-03

Tab – Project FTE Count

IMPORTING AND MODELING DATA

How does Power BI Import Data?

- Power BI is two products, Power Query and Power BI. You can also find Power Query in Microsoft Excel (and a lot of other places)
- When we import data, we do that through Power Query, we use this to manipulate our data (Microsoft uses a language called “M” in the background)
- Once we are ready, we then close Power Query which then “pushes” the data into Power BI. We can then manipulate the data further using a language called “DAX” (it is like creating formulas in Microsoft Excel)

IMPORTING AND MODELING DATA

What is M?

- M is the **data transformation engine** in Power BI
- M Query is a mashup query language used to query multiple sets of data sources
- M contains commands to transform data and can return the results of the query and transformations to the Power BI data model
- Normally, we use M Query to query data sources, clean and load data

IMPORTING AND MODELING DATA

What is M (continued)?

- You can use M for any data preparation and data transformation tasks before loading the data into your model
- For example, instead of bringing 3 tables into Power BI, you can remove unneeded columns and merge them all together using M and create a single table to load into the model
- This can reduce the load and improve performance once the data model is loaded

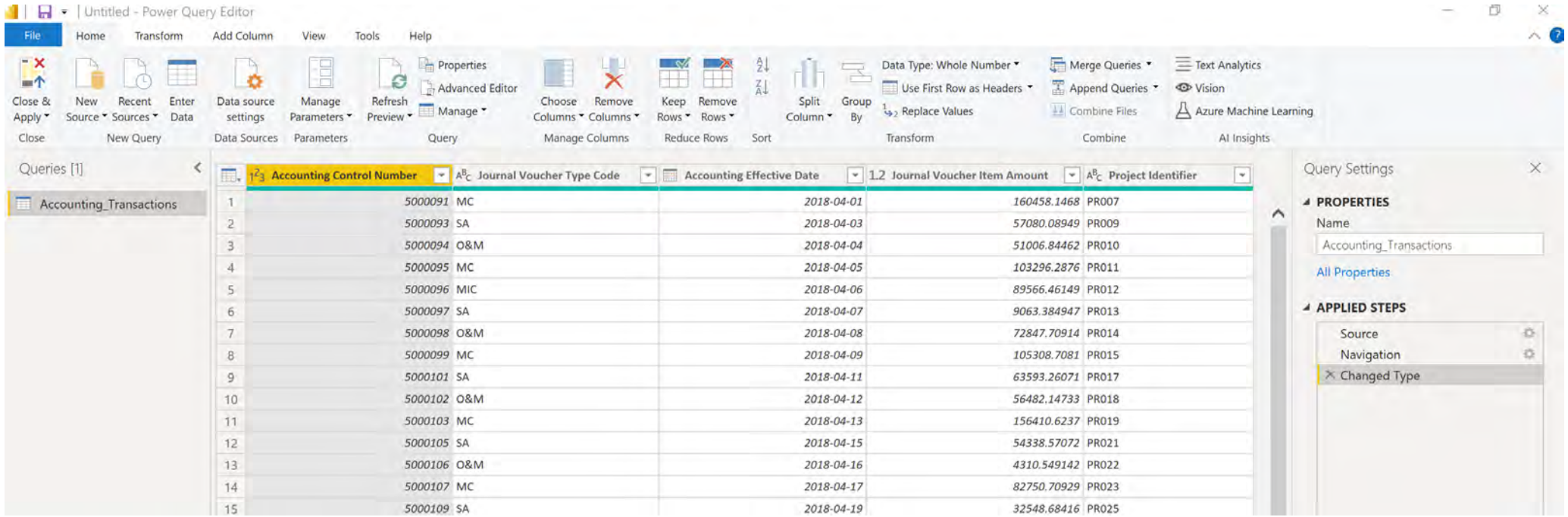
IMPORTING AND MODELING DATA

What is DAX ? We will come back to this in more detail later but for now...

- **D**ata **A**nalysis e**X**pression Language
- DAX is the common language used by SQL Server Analysis Services Tabular, Power BI, and Power Pivot in Excel
- DAX is the **analytical engine** in Power BI
- You use it once the data is loaded by creating custom columns, tables or measures, for instance
- Unlike M, it has some similarities to Excel functions
- However, DAX provides much more power than Excel formula (in many ways)

IMPORTING AND MODELING DATA

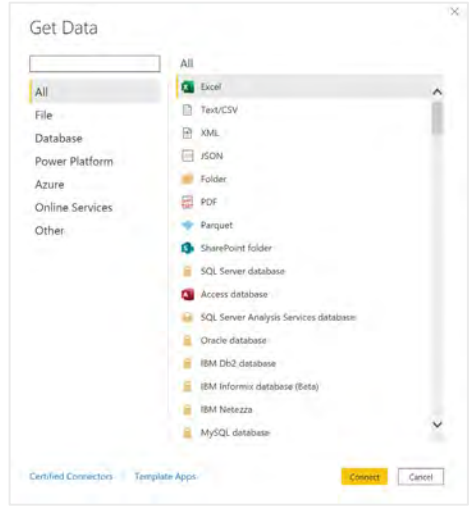
Typically, we want to “tweak” the data before loading it into Power BI. This can be done using the “Power Query” Interface.



IMPORTING AND MODELING DATA

Let's get some data into Power Query (follow the instructor):

1. Open Power BI.
2. Close the yellow "Hello" screen (we will come back to this later).
3. In the top "Home" tab you will see a "Get data" button. Click on that and the instructor will walk through several different options you have.
4. Once we have explored a few options select "Excel" .
5. Click on the file you saved called "Data Set - Accounting.xlsx" and click open.
6. PAUSE while the instructor tells you about all the different ways you can import the data (next slide).



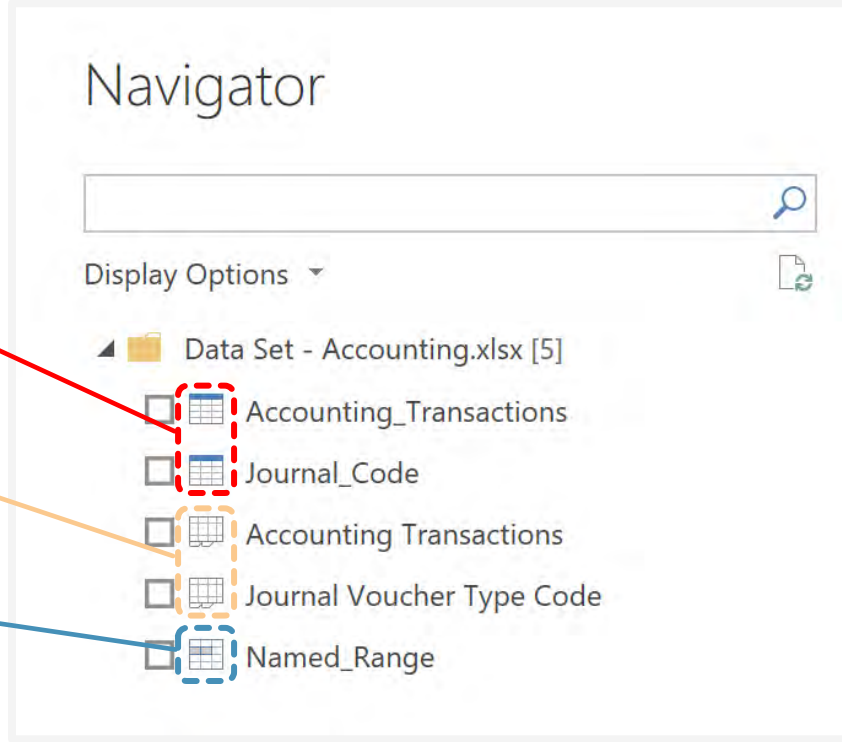
IMPORTING AND MODELING DATA

Here are the different options for importing data from an Excel spreadsheet:

These shapes represent a defined TABLE in the spreadsheet

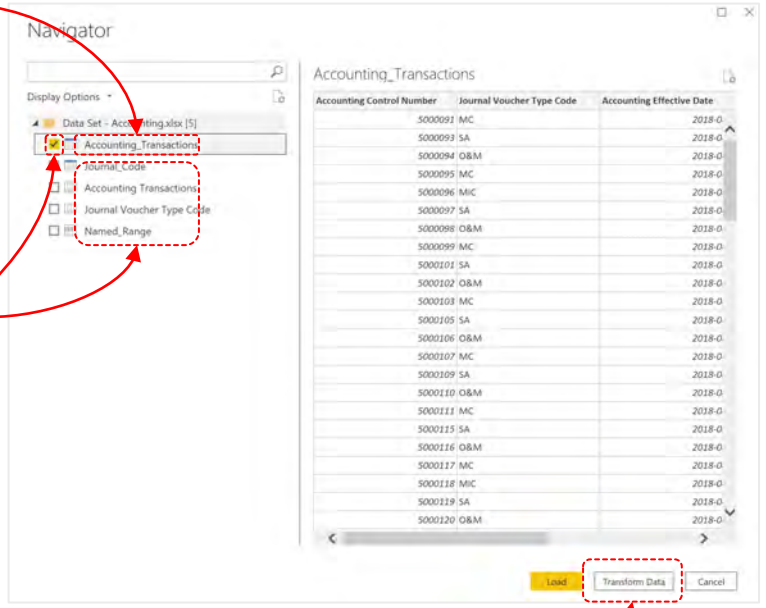
These shapes represent a defined TAB in the spreadsheet

This shape represents a defined NAMED RANGE in the spreadsheet



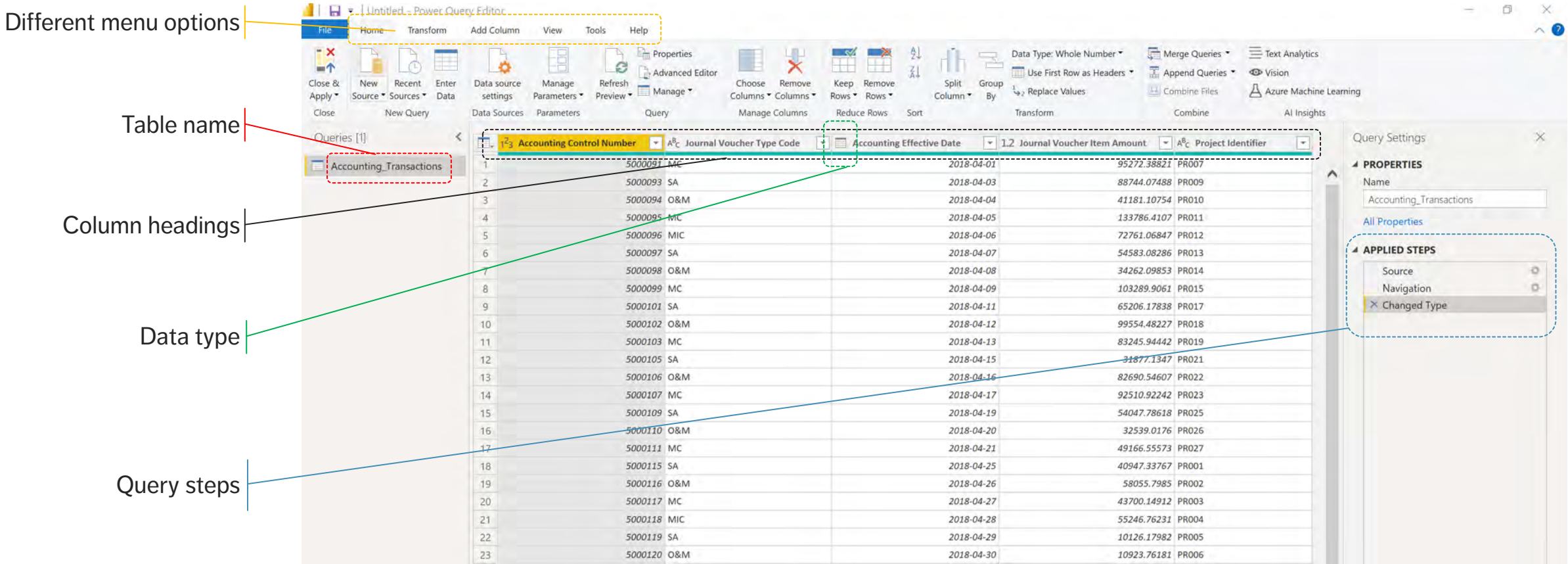
IMPORTING AND MODELING DATA

7. Click on the first TABLE named “Accounting_Transactions” but don’t select the check mark (yet).
8. You will see a summary of the data when you do.
9. Click on the other options to get a summary of that data.
10. Go back to “Acounting_Transactions” and select the checkbox.
11. **DON’T CLICK ON LOAD!!!!!**
12. Instead let's tweak the data a bit, so we will edit the transformation by clicking on “Transform Data”.



IMPORTING AND MODELING DATA

The instructor will walk you through the different parts of the Power Query screen:



IMPORTING AND MODELING DATA

Follow the instructor to do the following

1. Change the name of the table: Double click (or right click) on the table name and edit to remove the underscore.
2. Change the name of the “Accounting Effective Date” column to “Effective Date” and “Journal Voucher Item Amount” to “Item Amount” (note the new step in the query step box).
3. Remove the dates from March 2018 in the Effective Date Column: Click on the drop-down arrow by the column header, select “date filters”, then “after”, then select March 31 2018, then “ok”.
4. The instructor will show you where Power BI keeps the “M” language version of the query (this will be re-visited in a more advanced course).
5. Click on “Close and Apply” and the Instructor will take you through Power BI again but this time with data imported. Remember to save your pbix!

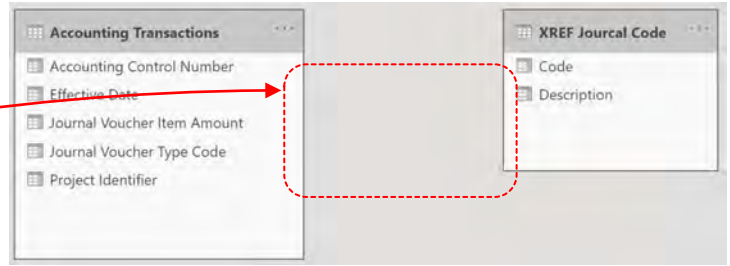
IMPORTING AND MODELING DATA

EXERCISE: go back to the same spreadsheet and add in the "Journal Voucher Type Code" tab. Remember to click on "Transform Data" to double check that all of the data looks good. Change the Table Name to "XREF Journal Code".

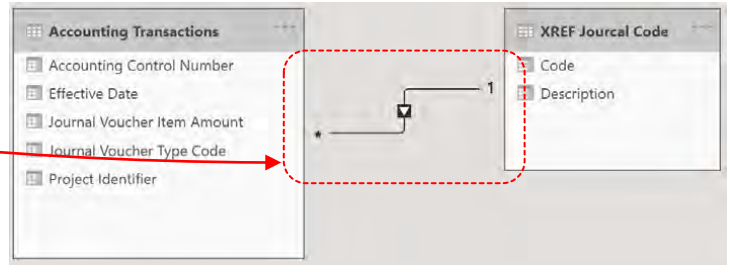
- XREF stands for "Cross Reference"
- Cross reference tables are tables that explain data and are not the primary data values we typically evaluate.
- By starting all your Cross Reference tables with XREF we can group them together in Power BI, this will help to keep you sane!

IMPORTING AND MODELING DATA

We now want to link our two tables together. Power BI is powerful because we approach data by considering it relations to other data (literally relational data)

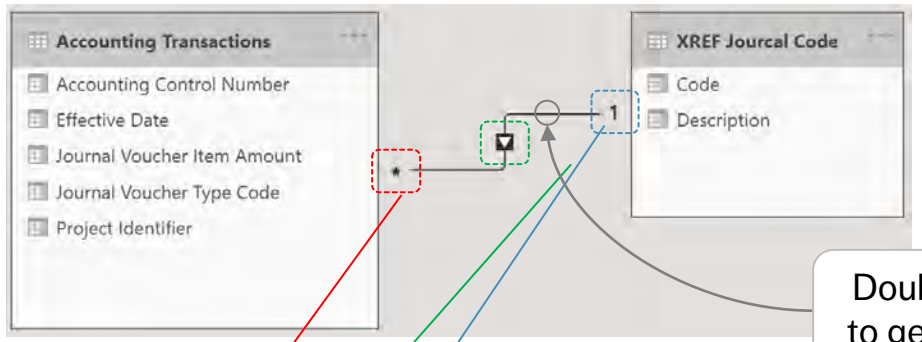


1. Click on the “Model” button in Power BI
2. You will see the two tables. but they are not yet linked
3. Follow the instructor to link the tables by doing the following:
    1. Left click on “XREF Journal Code – Code”
    2. Drag across to “Accounting Transactions – Journal Voucher Type Code”
    3. Release and see the new connection in our first data model!
4. Let’s go and take a close look…



IMPORTING AND MODELING DATA

Let's take a look at the connection between the two tables

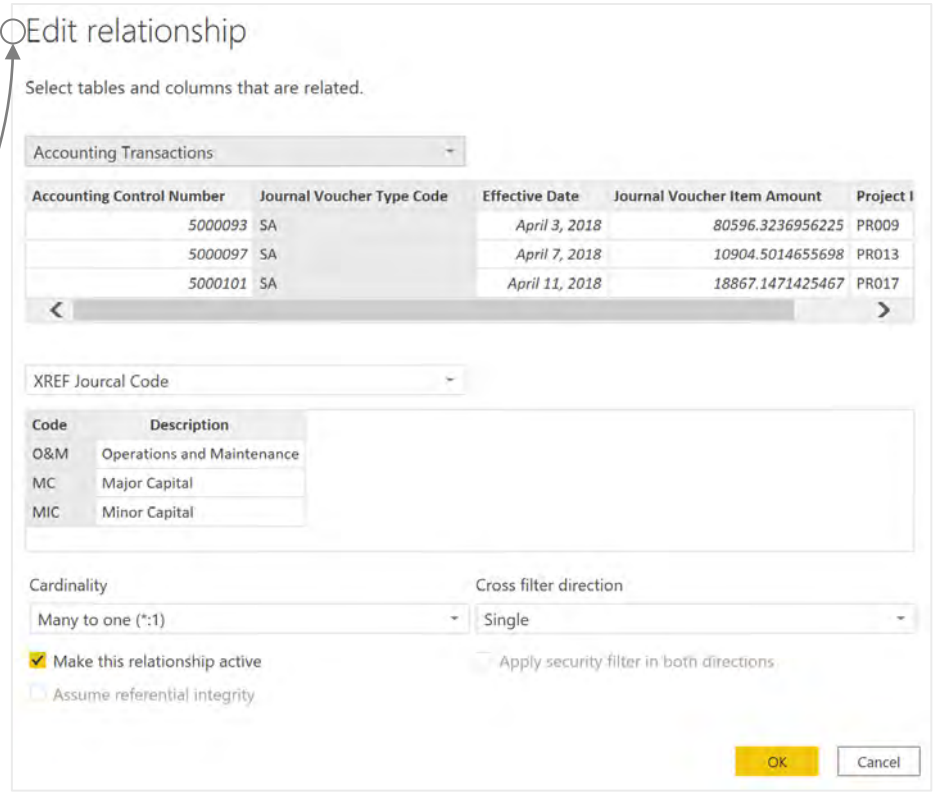


Double click the line to get this dialog box

Cardinality "Many"

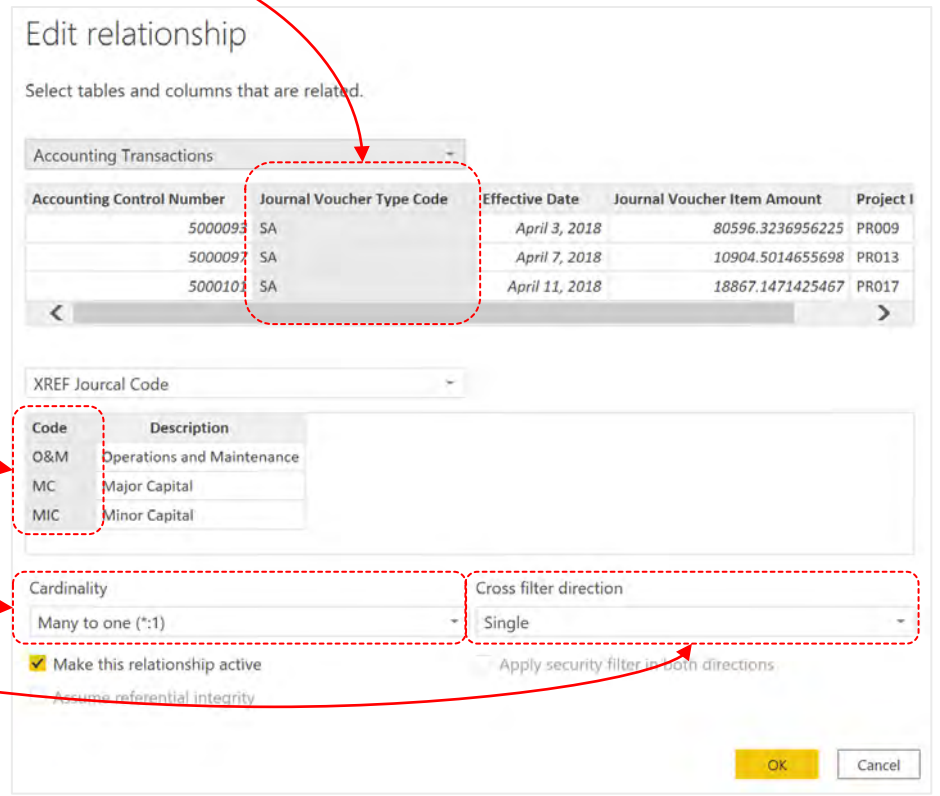
Relationship Direction

Cardinality "One"



IMPORTING AND MODELING DATA

- The greyed-out columns show the ones which are connected
- Cardinality describes how two tables are related.
  - One to many / many to one
  - One to one
  - Many to many (BAD)
- Cross filter direction – the direction in which Power BI applies a filter from one table to another. This is the Subject of a follow up course (always leave as single for now).



IMPORTING AND MODELING DATA

So, in summary we have done the following

1. Grabbed data from two separate tabs
2. Learned to ALWAYS take a first look at the data in Power Query
3. Changed a table name, column name, filtered a date and took a look at M in Advanced Editor
4. Closed the query and loaded the data into Power BI
5. Linked two tables together
6. Next it is time for us to do our first calculations and create our first charts!

CREATING OUR FIRST CALCULATIONS & CHARTS

PBI-1: GETTING TO KNOW POWER BI

CALCULATIONS – DAX

In Excel, formulas are entered in cells; in Power BI we work on independent calculations called measures, columns and tables instead, using a language called **DAX**.

DAX is a collection of

- Functions, operators, and constants
- Used to create new data entities – column, measures and tables

DAX helps create new information from data already found in the model.

Great resource:

- https://docs.microsoft.com/en-us/power-bi/guided-learning/introductiontodax

CALCULATIONS – DAX

This formula's syntax includes the following elements:

```
Total Amount = sum('Accounting Transactions'[Item Amount])
```

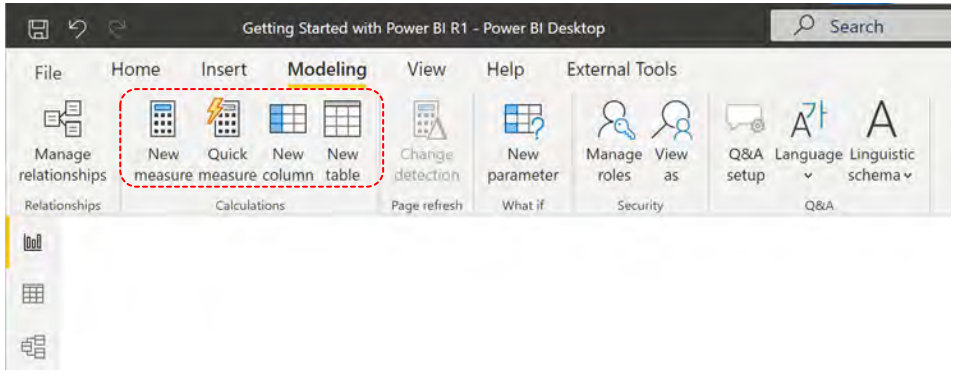
- The measure name is "`Total Amount`".
- The operator "`=`" indicates the beginning of the formula; when calculated, it returns a result.
- The function (in this case "`sum`") contains one argument which is contained in the parenthesis ( ).
- The table we are taking our data from is *before* the [ ] in this case it is '`Accounting Transactions`'.
- Note that as our table name contains two words is it contained in single quotations ' '. If our column name was one word it would not, i.e., `Accounting_Transactions[Item Amount]`.
- The referenced column from the table is inside the [ ] in this case it is `[Item Amount]`.

CALCULATIONS – COLUMNS VS. MEASURES VS. TABLES

Calculated Columns vs Measures vs Calculated Tables

In Excel we add a calculation to a cell. In Power BI a calculation can “live” in one of three places (not including in Power Query – more on that in another course). We can do calculations that create:

1. A new column of data, this is called a calculated column.
2. A new data value calculated solely to be used in a visualization, this is called a measure.
3. As new table of data, this is called a calculated table.



CALCULATIONS – COLUMNS VS. MEASURES VS. TABLES

Calculated Colums are:

- Additional columns of data added to an existing data table.
- Evaluated for each row in the table, immediately after 'Enter' is hit to complete the formula.
- Saved back into the model, therefore they take up space (if you have 1M rows of data and you do a calculated column you have just added 1M new data points.

CALCULATIONS – COLUMNS VS. MEASURES VS. TABLES

Calculated Measures are:

- Evaluated when used in a visual or when the visual is rendered.
- Therefore, they are not saved anywhere (aside from the cache).

Measures are generally used over calculated columns, but there are **trade-offs** with performance (report runtime vs. pre-processed), storage space, and the type of expressions that can be used.

Calculated columns are often used to **filter on the result** rather than just as a calculated result (for example in **slicers**).

CALCULATIONS – COLUMNS VS. MEASURES VS. TABLES

Calculated Tables are:

- New tables of data calculated from existing tables in your data model.
- Evaluated for each row in the source table, immediately after 'Enter' is hit to complete the formula.
- Like other tables they can have relationships with any other table in your data model.
- Saved back into the model, therefore they take up space.

CALCULATIONS – COLUMNS VS. MEASURES

When do we use calculated columns over measures? Sometimes either is an option, but in most situations **the computation needs determine the choice**.

A calculated column needs to be defined to do the following:

- place the calculated results in a slicer, or in the axis of a chart, or use the result as a filter condition in a DAX query;
- define an expression that is strictly bound to the current row (for example, “Cost * Volume” does not work on an average or on a sum of two or more columns);
- categorize text or numbers (for example, a range of values for a measure, a range of customer ages, such as 0–18, 18–25, etc.).

CALCULATIONS – COLUMNS VS. MEASURES

Typically, measures are defined whenever a resulting calculation needs to be displayed in the values area in the plot area of a chart:

- for instance, in calculating the cost percentage on a certain selection of data.
- Measures can use data from many different tables and therefore don't "belong" to a table in the same way a calculated column is.
- Because of this it is good practice to create a separate "home" for our measures.
- In general Measures are more computationally efficient that calculated columns and do not add additional data into our model.

CALCULATIONS – CALCULATED TABLES

We are not looking at Calculated Tables as part of this course, but they are typically used when:

- The result you are calculating is a matrix of some description.
- You need to summarize data from other tables.
- You need to isolate data from other tables.
- You need to create a new data relationship in your data model.

CALCULATIONS – ARITHMETIC FUNCTIONS

In a previous slide we showed a standard use of SUM; DAX has the expected standard arithmetic expressions (SUM, DIVIDE, PRODUCT, MAX, MIN, AVERAGE, etc.)

Most arithmetic expressions are straightforward, but some are very specific, for example Divide. Don't give in to the temptation to manually divide using a "/" operation, say (although everybody does and sometimes it is ok) – there is a failure option built into (for example) DIVIDE (or an alternate result, such as "0" for a "divide by 0" error).

```
DIVIDE(<numerator>, <denominator> [,<alternateresult>])
```

Spend a few minutes on a new tab playing around with these expressions; we will see them in action in a short while.

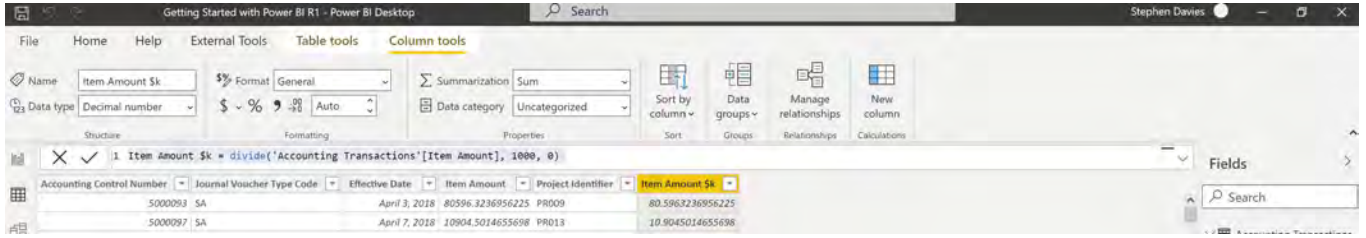
CALCULATIONS – OUR FIRST CALCULATED COLUMN

Let's build our first DAX expression, we are going to divide our "Item Amount" column by 1000 to have a new column that represents the $ amount in $k.

To do this follow the instructions below:

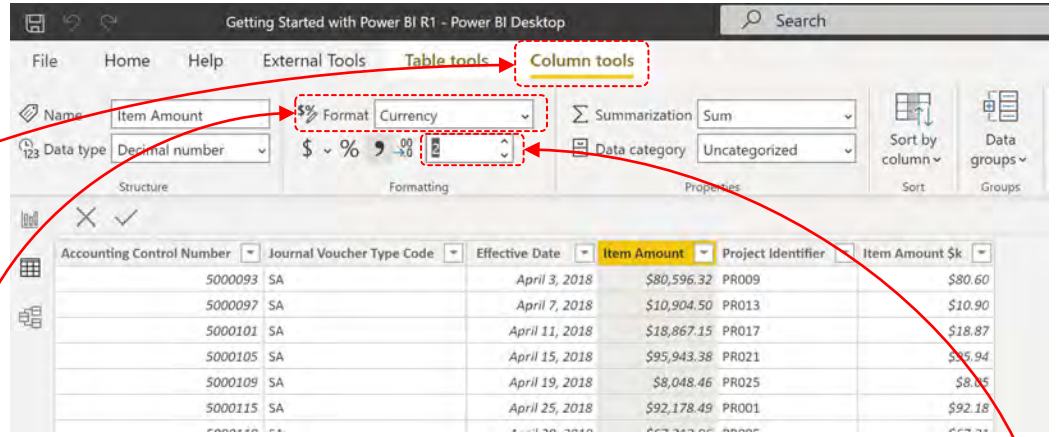
1. Click on the table "Accounting Transactions" you will now see a new menu item "Table tools"
2. Click on "New Column" (you can access this from "Home" or "Table tools"
3. Type in the formula then hit Enter:

```
Item Amount $k = divide('Accounting Transactions'[Item Amount], 1000, 0)
```



CALCULATIONS – OUR FIRST CALCULATED COLUMN

Now we have our first new piece of data let's get it into a chart. Before we do that let's do a couple of housekeeping items (follow the instructor):

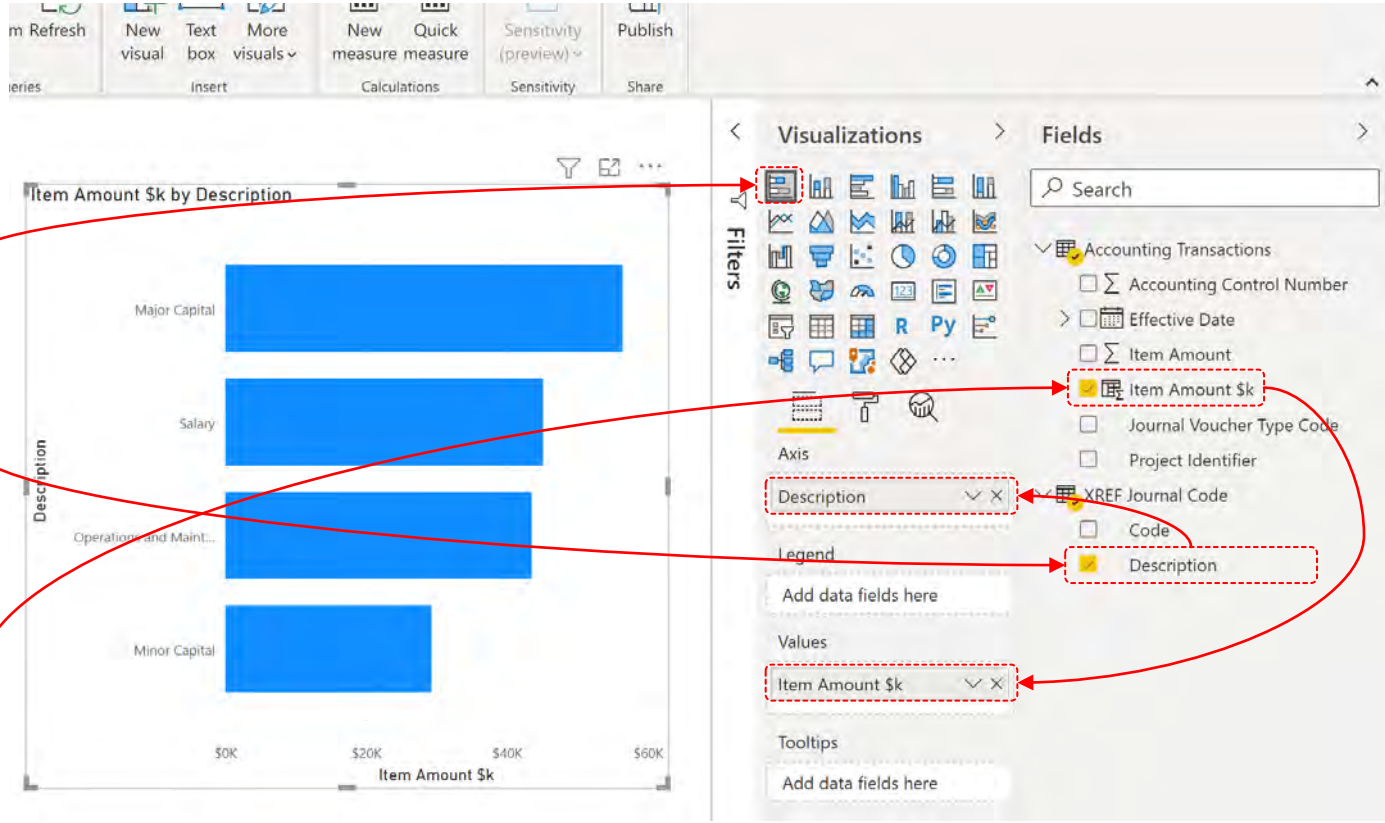


1. Click on the new column (Item Amount $k), the menu item "Column tools" will appear in the top menu
2. In the "Formatting section" of the menu there is a "Format" dropdown. Click on that and change it to Currency
3. Let's change to 2 decimal places. Again, in the formatting section of the menu, it will have a box that says "Auto". Click on that and change it to "2"
4. Repeat for the original "Item Amount" column

CALCULATIONS – OUR FIRST CALCULATED COLUMN

Let's make a quick chart using this new column (follow the instructor).

1. Go to the "Report" Screen
2. Under Visualizations click on "Stacked Bar Chart"
3. From the "XREF Journal Code" table drag "Description" column on the "Axis" field
4. Finally, from the "Accounting Transactions" table drag the "Item Amount $k" column onto the "Values" field



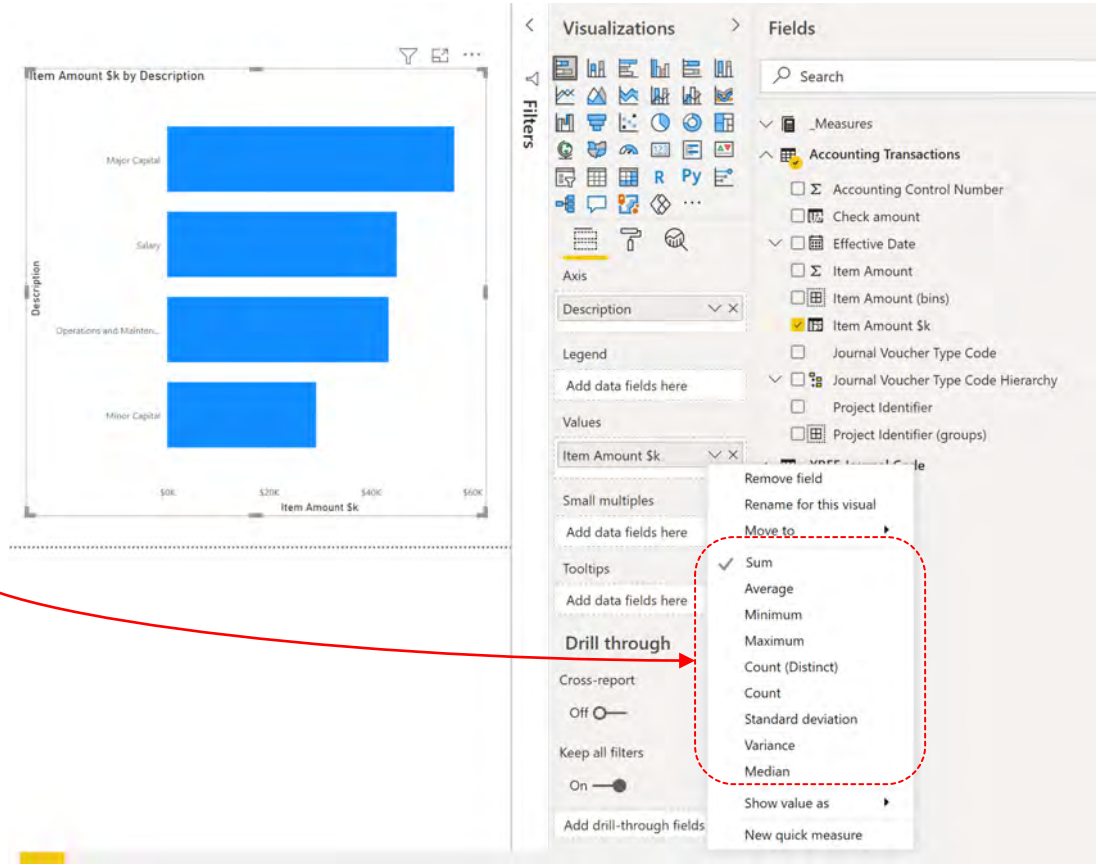
CALCULATIONS – OUR FIRST CALCULATED COLUMN

Functionality Alert!!!

When we drag a field onto the “Values” section we have several options.

Power BI defaults to “sum” but we have a lot more options especially for numerical columns.

For Text we only are able to “Count” and Count (Distinct)”.



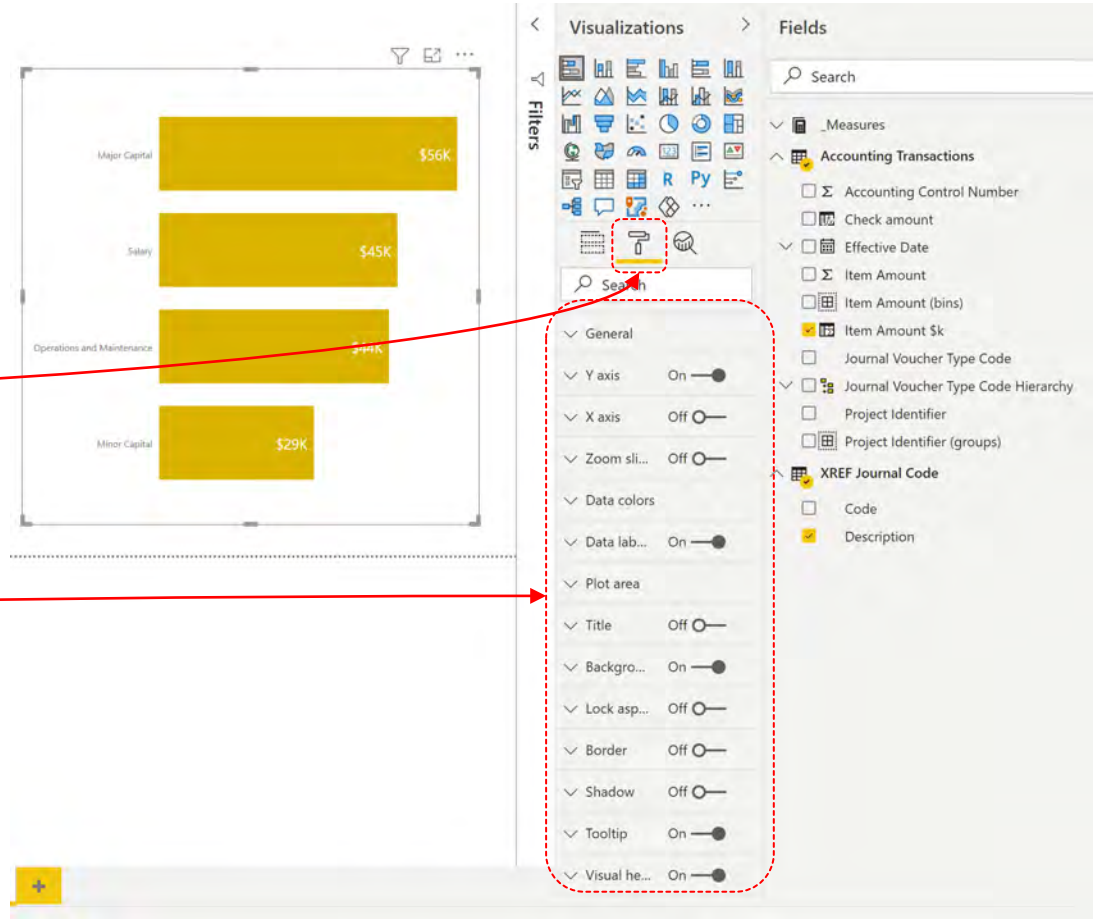
CALCULATIONS – OUR FIRST CALCULATED COLUMN

Functionality Alert!!!

We might also want to do some formatting on our chart.

To do this click on the chart and then click on the button that looks like a paint roller.

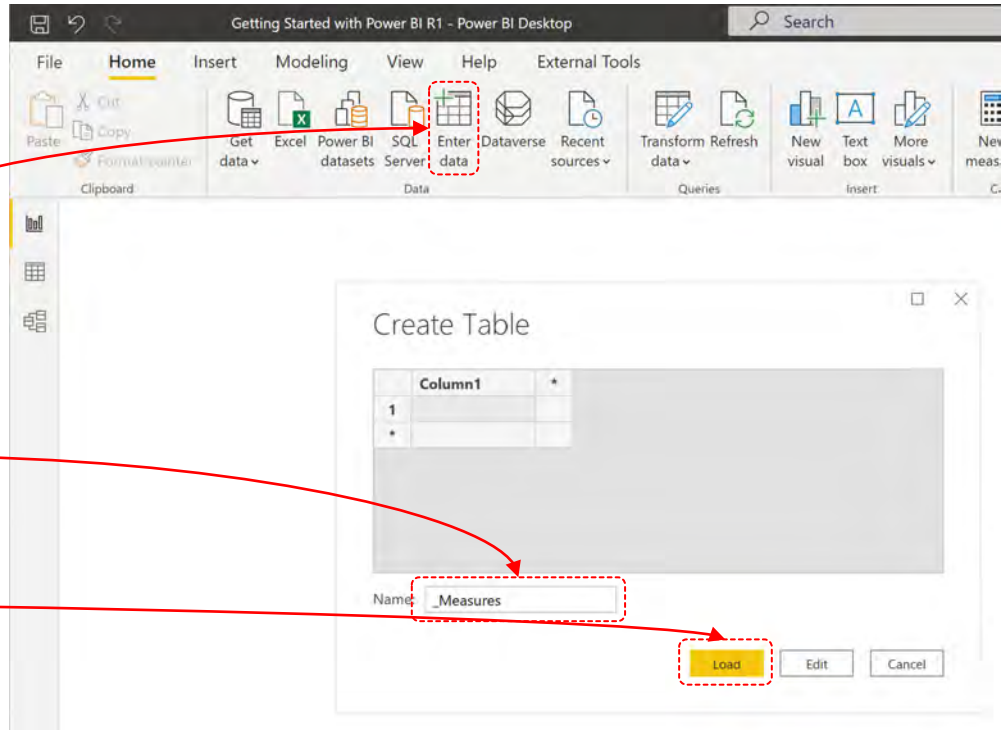
Your instructor will walk you through some of the basic formatting options (they change depending on the type chart that you select).



CALCULATIONS – OUR FIRST MEASURE

We are now going to build our first measure. Before we do that, we should follow some best practice and create a "Measures Table" for our measure to live in:

1. In the "Home" Menu, in the "Data" section click on "Enter Data".
2. Ignore everything excel the "name" at the bottom.
3. Override the name "Table" with "_Measures" (**make sure you include the underscore**).
4. Click on "Load".
5. You will now see the table appear at the top (because of the underscore).

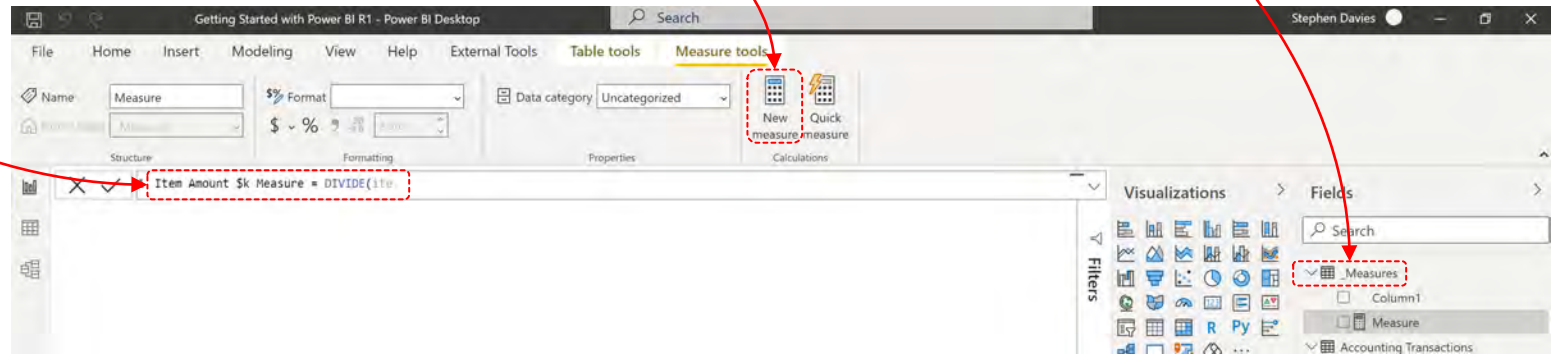


CALCULATIONS – OUR FIRST MEASURE

Next, we are going to create a measure using the “Divide” function again. To create the measure do the following:

1. Click on the new “_Measures” table
2. The top menu will change to “Table tools”
3. Click on “New measure”
4. Start to TYPE the following (don’t cut and paste): `Item Amount $k Measure = divide(item`

… hang on, Power BI is not letting us complete this in the same way that we did for a calculated column – why?



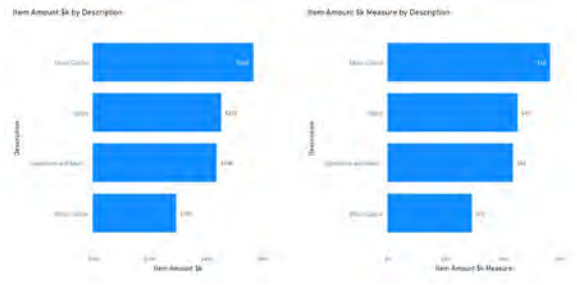
CALCULATIONS – OUR FIRST MEASURE

It's because a measure requires us to *perform an operation* on a column before we can divide i.e.:

```
Item Amount $k Measure = DIVIDE(sum('Accounting Transactions'[Item Amount $k]), 1000, 0)
```

By default, measures perform an operation on the entire column, not row by row (there are exceptions which are covered off in another course).

If we wanted to, we could DIVIDE the average, max, min, standard deviation etc.



Complete the measure using the formula above and then create a chart as we did for the calculated column.

NOTE: even though we created the measure from a column that was formatted as currency, we still need to format the Measure as currency separately.

CALCULATIONS – FILTERING & SLICING

As we can see in our previous examples, we are performing an operation on an entire column of data, but what if we want to only do that on a subset of the data.

There are a number of ways to do this but in Power BI we typically:

- Build the filter into the calculation itself.
- Filter the result using a slicer (more on that later) or using the Power BI built in filter function.

We are going to do an example of each.

CALCULATIONS – FILTERING & SLICING

First, we are going to build a filter directly into our calculations, we are going to do this using probably THE most useful DAX operation – “Calculate”

Let’s do an example, suppose we want to sum the $ amount but just for Salaries (in the Accounting Transactions table), we would use Calculate to do that in the following way:

1. Click on our new “_Measures” table
2. Type in the following

```
Total Salary = CALCULATE(sum('Accounting Transactions'[Item Amount]),
                         'Accounting Transactions'[Journal Voucher Type
                             Code]="SA")
```

This is the operation that is being calculated

This is the filter that is being applied

CALCULATIONS – FILTERING & SLICING

Best Practice Alert!!

As a quick aside in the previous slide notice that there is a carriage return after the first line. This is done in Power BI by holding down “Shift” and hitting “Enter” (your instructor will show you).

It is good practice to do this to make sure that your measures and columns are easy to read.

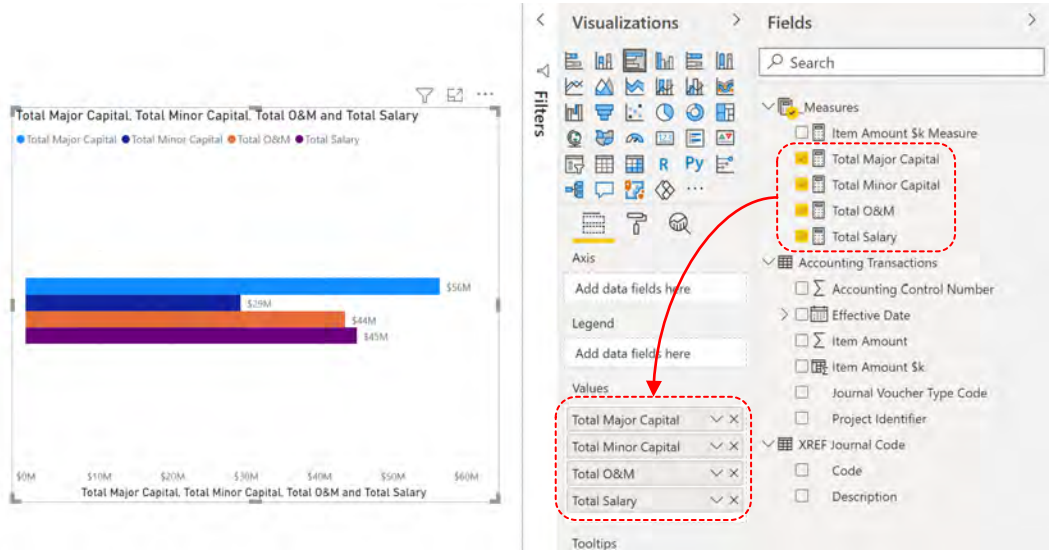
Another good piece of best practice is to comment your code. You can do this by typing in 2 forward slashes, see the example below:

```
Total Salary = CALCULATE(sum('Accounting Transactions'[Item Amount]), //this is a comment
        'Accounting Transactions'[Journal Voucher Type Code]="SA") //another comment
```

CALCULATIONS – FILTERING & SLICING

Exercise: create three more measures in the measures table, one for each of the Journal Voucher Type Codes – MC, MIC, O&M, SA (already done). Remember to format each measure as you go.

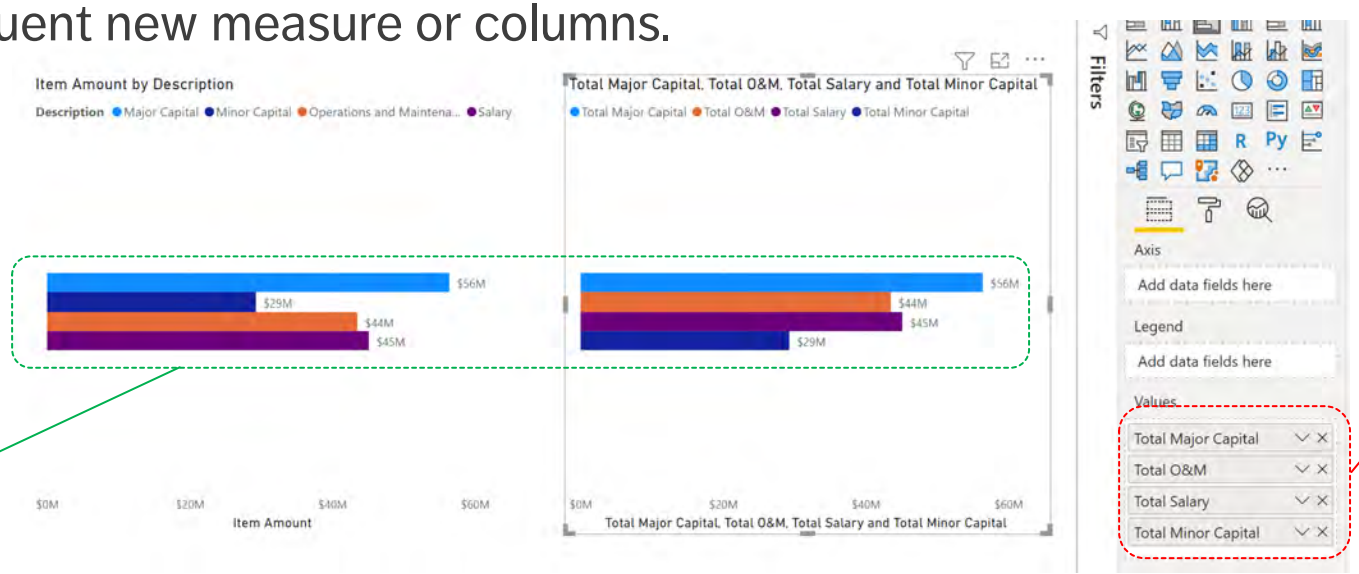
Once you are done, insert a "Clustered Bar Chart" and drag each new measures into the Values space.



CALCULATIONS – FILTERING & SLICING

But didn't we achieve the same result by using "Journal Voucher Type Code" (or the Description in our XREF table) by either dragging that column onto "Axis" or "Legend".

We can do that but the nice thing about doing it as measures is that it is easy to re-order and we can use them for subsequent new measure or columns.



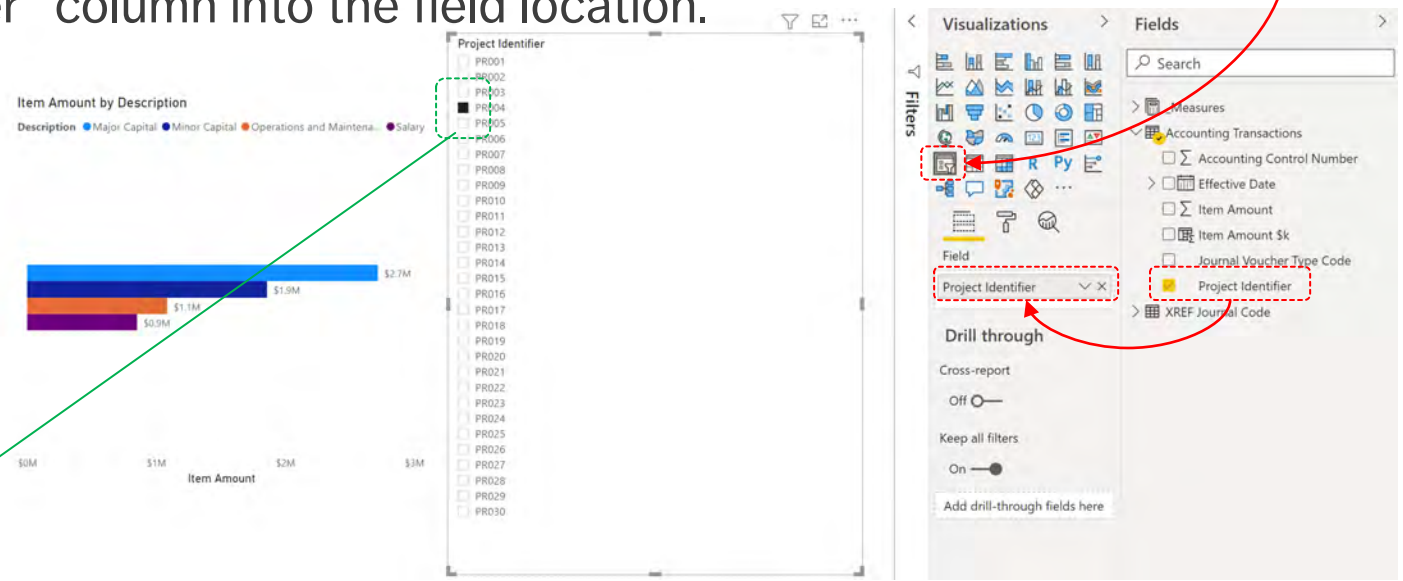
Different order of columns

Simply drag to re-order

CALCULATIONS – FILTERING & SLICING

We can also filter our visualizations by using slicers. In fact, Power BI by default allows a user to click on any chart which will in turn filter other charts.

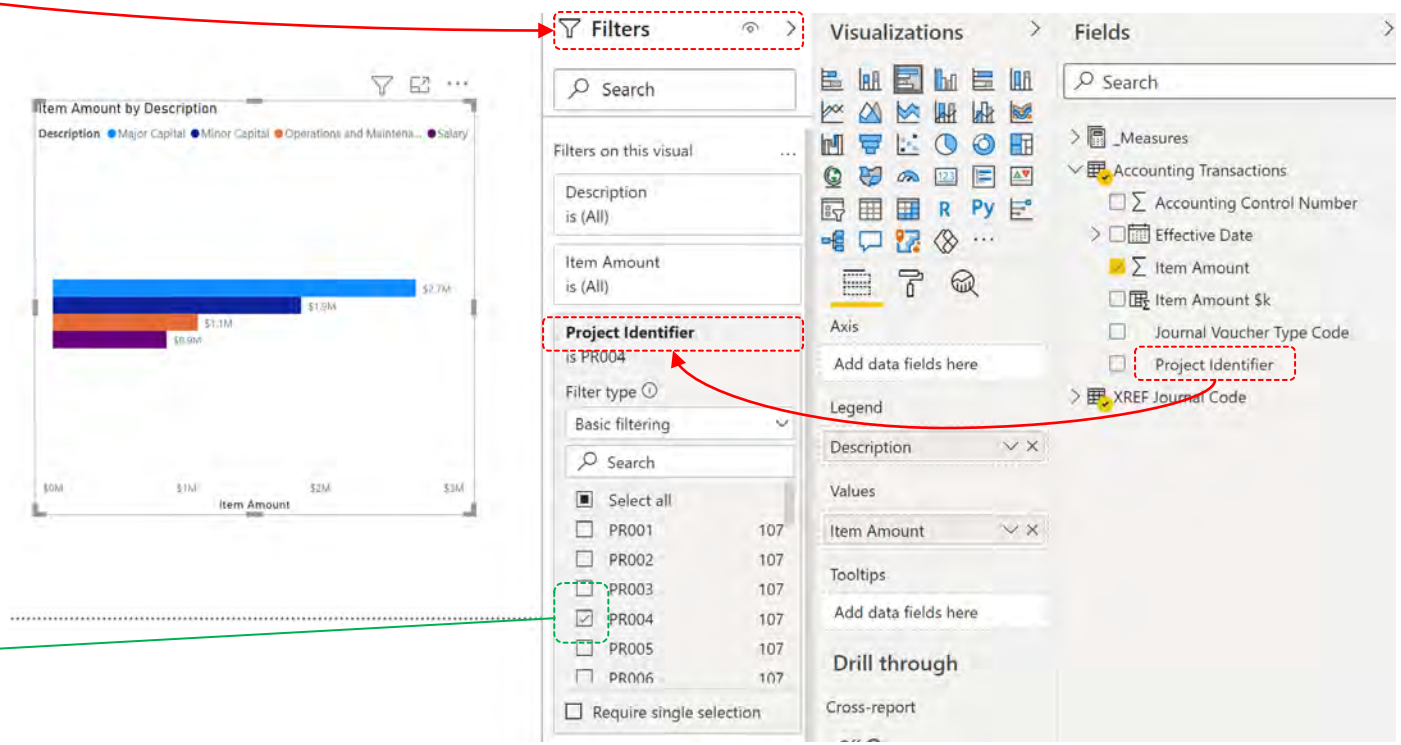
Let's setup a simple slicer using Project Identifier. Click on the slider icon to insert and then drag the "Project Identifier" column into the field location.



Select a value to filter the chart

CALCULATIONS – FILTERING & SLICING

We can also filter our visualizations by using the filter pane: click on a visualization and expand the filter pane, then select the column you wish to filter (or add another column) and select the value.



Select a value to filter the chart

CALCULATIONS – LOGICAL OPERATIONS

We are going to finish off our DAX by trying a logical operation. This is where we apply some Boolean logic like an “AND” or an “OR” operation.

There are MANY ways of doing this, but we are going to try a simple “IF” statement that selects a value if a logical operation is met.

Let us take a simple scenario where we want to flag an Item Amount over $100,000. We want to flag it so that the value can be checked. To do this we are going to add an additional column that has the following logic:

- If the value is equal to or greater then $100,000 then add the word “Check”.
- If the value is less that $100,000 then we leave the value in the new column blank.

CALCULATIONS – LOGICAL OPERATIONS

Follow the instructor through these steps:

1. Click on the “Accounting Transactions” table
2. Click on “Column Tools”, then “New Column”
3. Type in the following formula:

```
Check amount = if('Accounting Transactions'[Item Amount] >= 100000, // logical test
    "Check", // value if true
    blank()) // value if false
```

The “if” statement is the same as what we use in Excel

The value that is returned if the test is true

We cannot use “” to return a blank cell as Power BI interprets that as text. We should always use the “Blank()” function.

CALCULATIONS – ADDITIONAL EXERCISES

Try the following as exercises:

1. Try to create a new measure using existing measures. Try to create a measure that show the ratio of Minor to Major Capital.
2. Create a new LINE chart using this measure and for the X axis use “Effective Date”.
3. Create a new slicer that uses “Effective Date” and try it out to see how it changes the chart.
4. Add in one of the previous charts we created (or any new one) and click on it to see how it changes the other charts on the report.

ORGANIZING DATA

PBI-1: GETTING TO KNOW POWER BI

ORGANIZING DATA

Organizing our data is a critical part of building reports and dashboards. Fortunately, Power BI has some powerful built-in capabilities.

In this session we are going to look at the following:

- Built-in and custom Hierarchies
- Grouping numerical data
- Grouping text data
- Custom sorting of charts

ORGANIZING DATA – HIERARCHIES

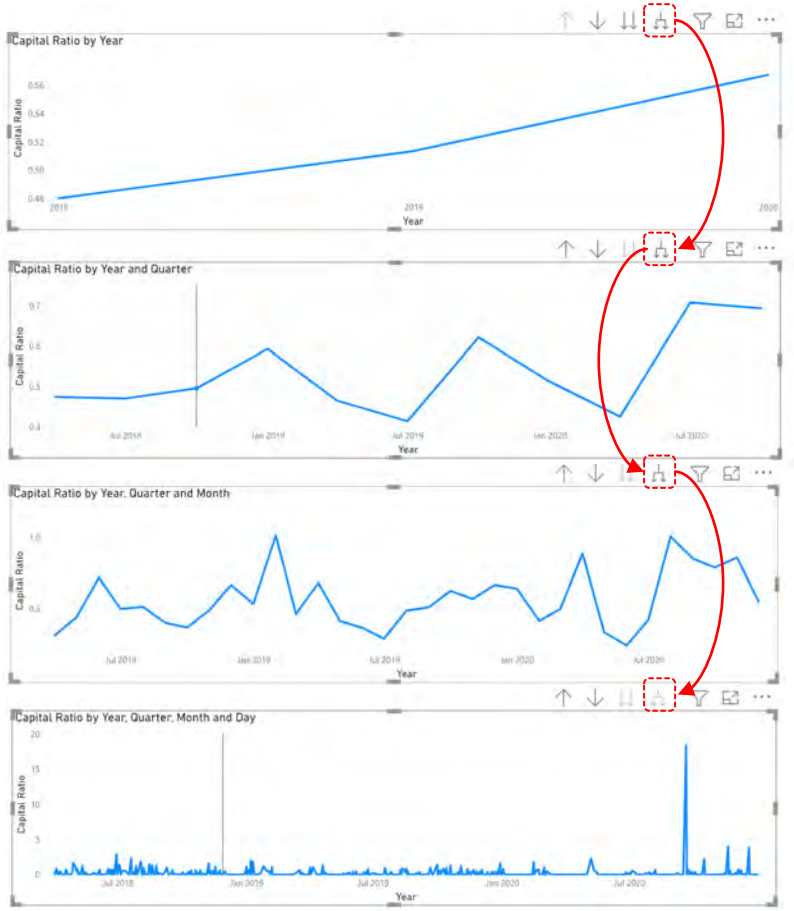
Hierarchies provide a way to order data levels. They let us **summarize** the data in different ways.

When you import a date, Power BI automatically creates a Date hierarchy, specifically:

- Year > Quarter > Month > Week > Day

The hierarchies can be navigated up and down in the line chart we created in the final exercise in the previous section.

We can also create custom Hierarchies as well.

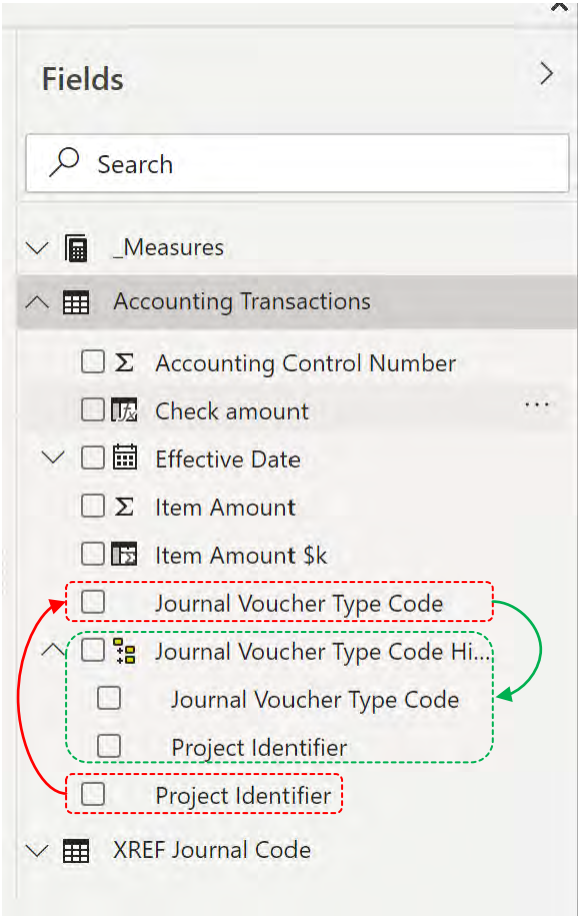


ORGANIZING DATA – HIERARCHIES

It is a very simple process to create a custom Hierarchy in Power BI. You simply drag one column on top of another.

The instructor will walk you through the following steps:

1. Make sure you are in the Report View.
2. Left click on “Project Identifier”.
3. Drag it over “Journal Voucher Type Code” and release – you should see the new Hierarchy.
4. Create a bar chart and add the Hierarchy to the Axis and “Item Amount” to the values.



ORGANIZING DATA – HIERARCHIES

Hierarchies are also great in Matrices.

Follow the instructor with the following steps:

1. Insert a Matrix.
2. Add the new Hierarchy to the “Rows” section.
3. Add “Item Amount” to the “Values” section.
4. Open the “Effective Date” Hierarchy and drag in just the “Year” to the “Columns” section.
5. Voila, this is typically what we do when we build Excel Pivot Tables!!

Journal Voucher Type Code	2018	2019	2020	Total
⊞ MC	$17,572,864.87	$22,225,788.16	$16,625,119.92	$56,423,772.95
⊟ MIC	$8,439,250.58	$11,414,651.88	$9,434,634.43	$29,288,536.89
PR001	$315,114.47	$416,740.47	$410,872.18	$1,142,727.12
PR002	$311,027.94	$865,044.52	$344,025.89	$1,520,098.35
PR003	$377,255.99	$376,393.65	$300,763.59	$1,054,413.23
PR004	$804,807.85	$650,695.17	$403,484.43	$1,858,987.45
PR005	$257,036.10	$504,146.07	$622,410.04	$1,383,592.21
PR006	$298,617.80	$846,093.69	$286,510.12	$1,431,221.61
PR007	$251,982.01	$635,896.07	$451,812.08	$1,339,690.16
PR009	$410,736.12	$686,732.38	$688,870.98	$1,786,339.48
PR010	$398,111.09	$840,385.74	$366,744.60	$1,605,241.43
PR011	$550,613.69	$524,861.77	$547,366.47	$1,622,841.92
PR012	$893,262.79	$543,928.48	$508,765.50	$1,945,956.77
PR013	$510,001.62	$515,346.14	$737,165.58	$1,762,513.34
PR015	$387,501.79	$509,568.68	$513,168.38	$1,410,238.84
PR017	$329,647.26	$578,441.38	$838,243.02	$1,746,331.66
PR018	$805,048.06	$581,937.69	$566,901.04	$1,953,886.78
PR019	$396,951.18	$583,673.99	$537,099.11	$1,517,724.29
PR022	$356,391.41	$778,093.13	$577,857.84	$1,712,342.38
PR023	$445,441.39	$619,313.93	$468,989.34	$1,533,744.66
PR027	$339,702.05	$357,358.91	$263,584.25	$960,645.21
⊞ O&M	$13,209,964.54	$16,233,373.54	$14,086,890.67	$43,530,228.74
⊞ SA	$13,741,811.83	$17,527,476.82	$13,860,358.05	$45,129,646.70
Total	$52,963,891.82	$67,401,290.39	$54,007,003.07	$174,372,185.28

ORGANIZING DATA – GROUPS

Groups are when we want to categorize or classify similar rows of data together in a table.

In Power BI there is a built-in function to do this for two types of data:

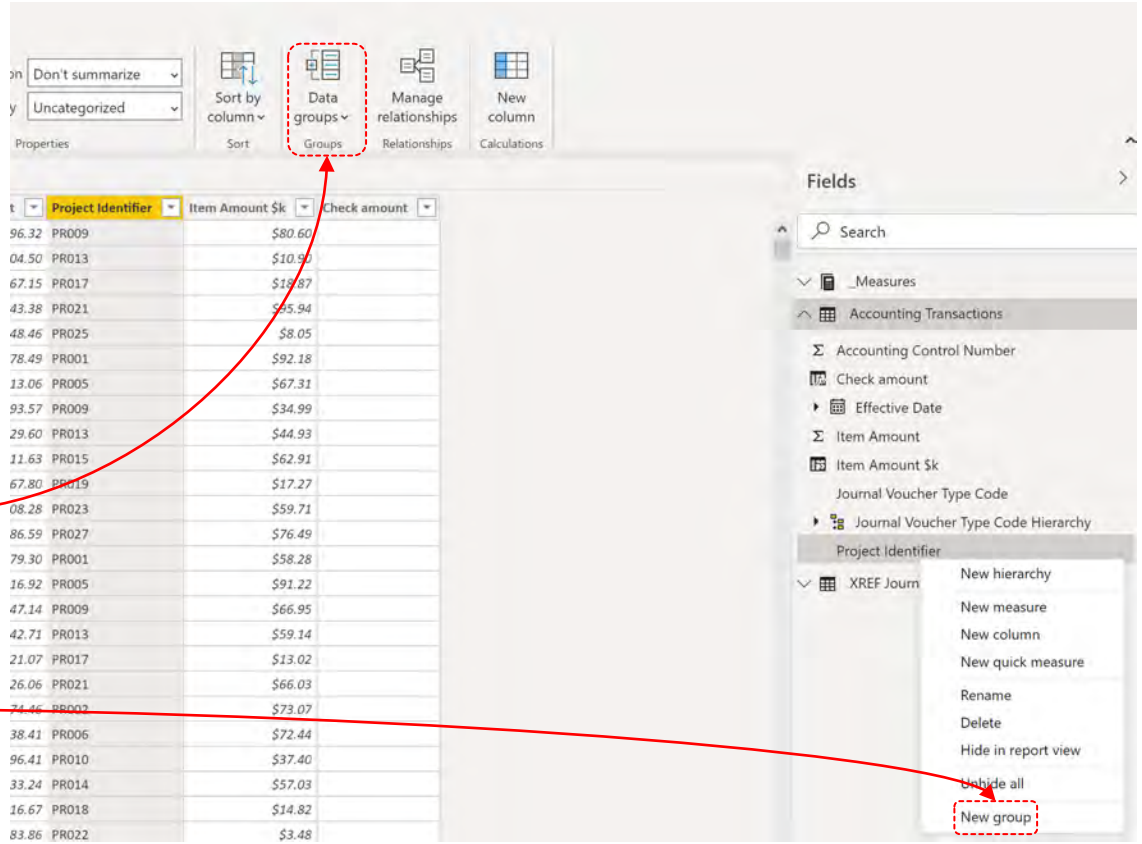
1. Grouping text together, e.g., grouping all EX equivalent job categories together
2. Grouping or “Bucketing” numbers, e.g., for creating an axis where we can count the number of items that fall in a particular set of ranges (0-100, 101-200, 201-300, > 300)

Neither approach is perfect but will cover off 90% of a new users' requirements. More complex scenarios are dealt with in one of our Power BI Intermediate courses

ORGANIZING DATA – GROUPS (TEXT)

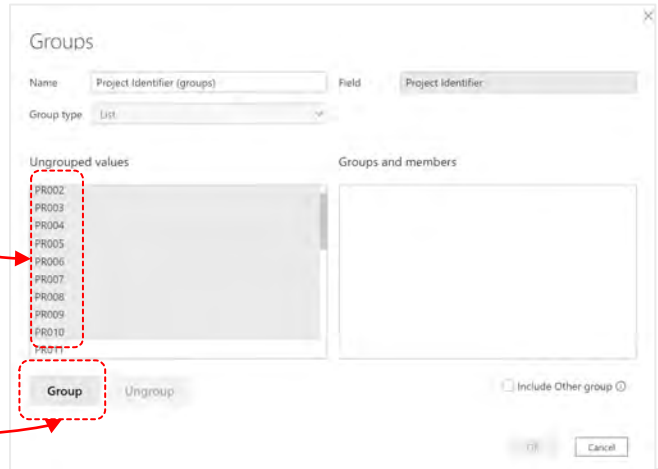
To work through an example of grouping text follow the instructor through the following steps:

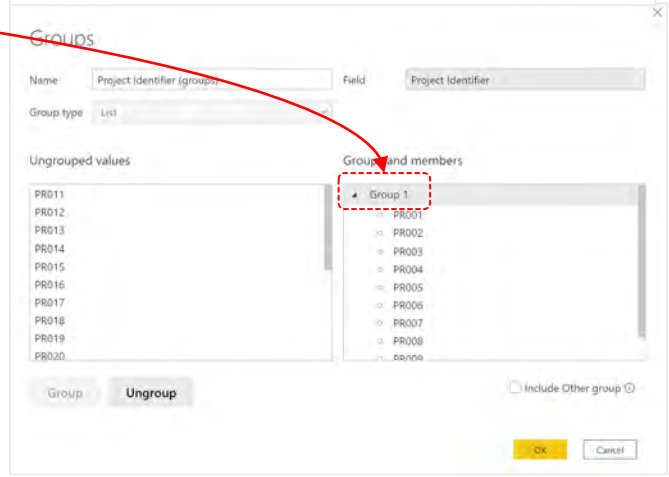
1. In either the Data or Report view and click on the “Accounting Transactions” Table.
2. Click on the “Project Identifier” column and either click on “Data groups” in the “Column Tools” menu or the three dots next to the “Project Identifier” name in the Fields menu and select “New group”.

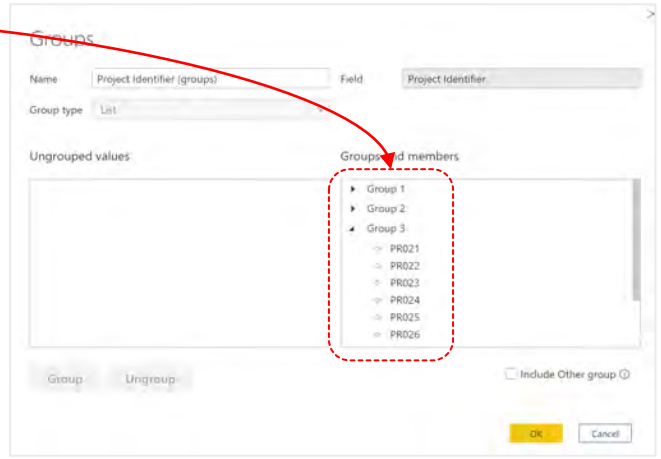


ORGANIZING DATA – GROUPS (TEXT)

3. You will then see a field called “Ungrouped Values”.
4. Click on PR001, hold down SHIFT and click on PR0010 to select that block (CTRL also works for individual items).
5. Click on “Group” and rename the Group in the “Groups and members” field to “Group 1”.
6. Repeat the process for PR0011 – PR0020 (Group 2) and PR0021 – PR0030 (Group 3).
7. Hit “OK”.







ORGANIZING DATA – GROUPS (TEXT)

You will now see a new column in the table with the Group associated with that row.

This column can be used in a chart axis, slicer, filter or as data (for count operations only).

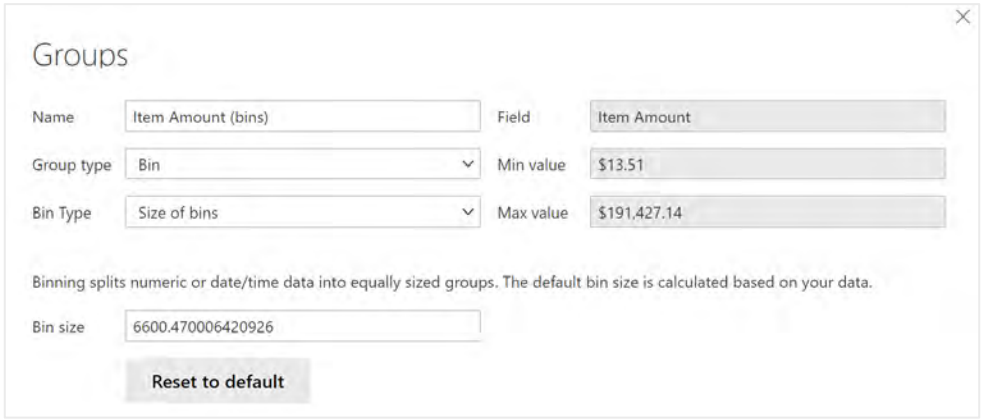
NOTE that if you want am “Other bucket” then check the “Include other group” checkbox in the Groups dialog box.

Accounting Control Number	Journal Voucher Type Code	Effective Date	Item Amount	Project Identifier	Item Amount $k	Check amount	Project Identifier (groups)
5000093	SA	April 3, 2018	$80,596.32	PR009	$80.60		Group 1
5000097	SA	April 7, 2018	$10,904.50	PR013	$10.90		Group 2
5000101	SA	April 11, 2018	$18,867.15	PR017	$18.87		Group 2
5000105	SA	April 15, 2018	$95,943.38	PR021	$95.94		Group 3
5000109	SA	April 19, 2018	$8,048.46	PR025	$8.05		Group 3
5000115	SA	April 25, 2018	$92,178.49	PR001	$92.18		Group 1
5000119	SA	April 29, 2018	$67,313.06	PR005	$67.31		Group 1
5000123	SA	May 3, 2018	$34,993.57	PR009	$34.99		Group 1
5000127	SA	May 7, 2018	$44,929.60	PR013	$44.93		Group 2

ORGANIZING DATA – GROUPS (NUMBERS)

To work through an example of grouping numbers follow the instructor through the following steps:

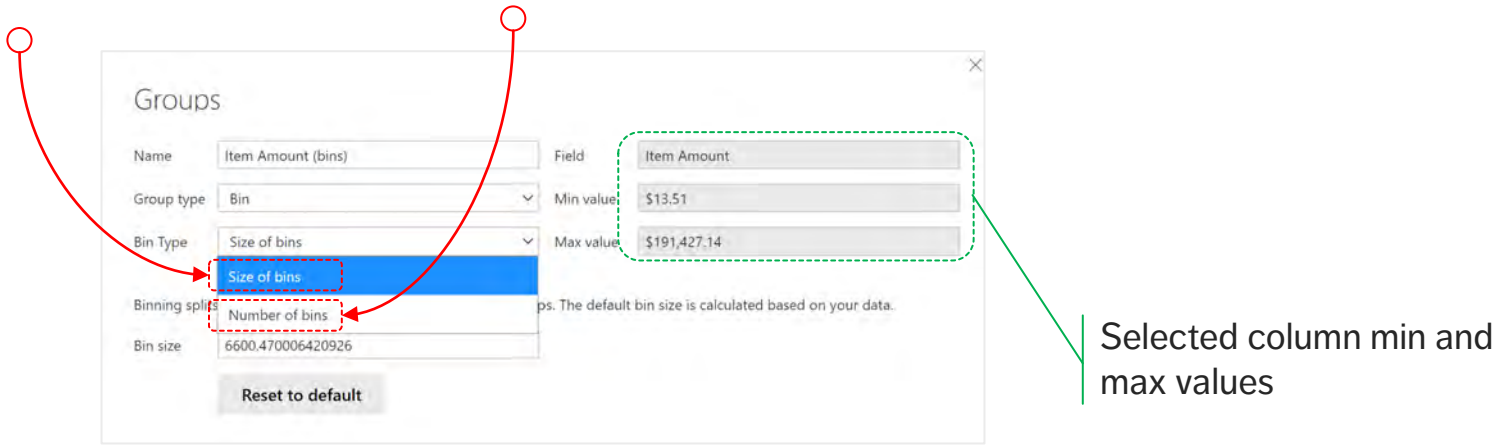
1. In either the Data or Report view and click on the “Accounting Transactions” Table
2. Click on the “Item Amount” column and as per the previous example either click on “Data groups” in the “Column Tools” menu or the three dots next to the “Item Amount” name in the Fields menu and select “New group”, you will see the following dialog:



ORGANIZING DATA – GROUPS (NUMBERS)

There are 2 ways of Grouping numerical data which can be selected using the “Group Type” option in the previous dialog

1. Lists – this is the same as the text example we worked through and behaves in the same way.
2. Bins – this is where we can create “buckets” to categorize the numbers in the column. There are 2 ways of doing this: “Size of Bins and “Number of Bins”.

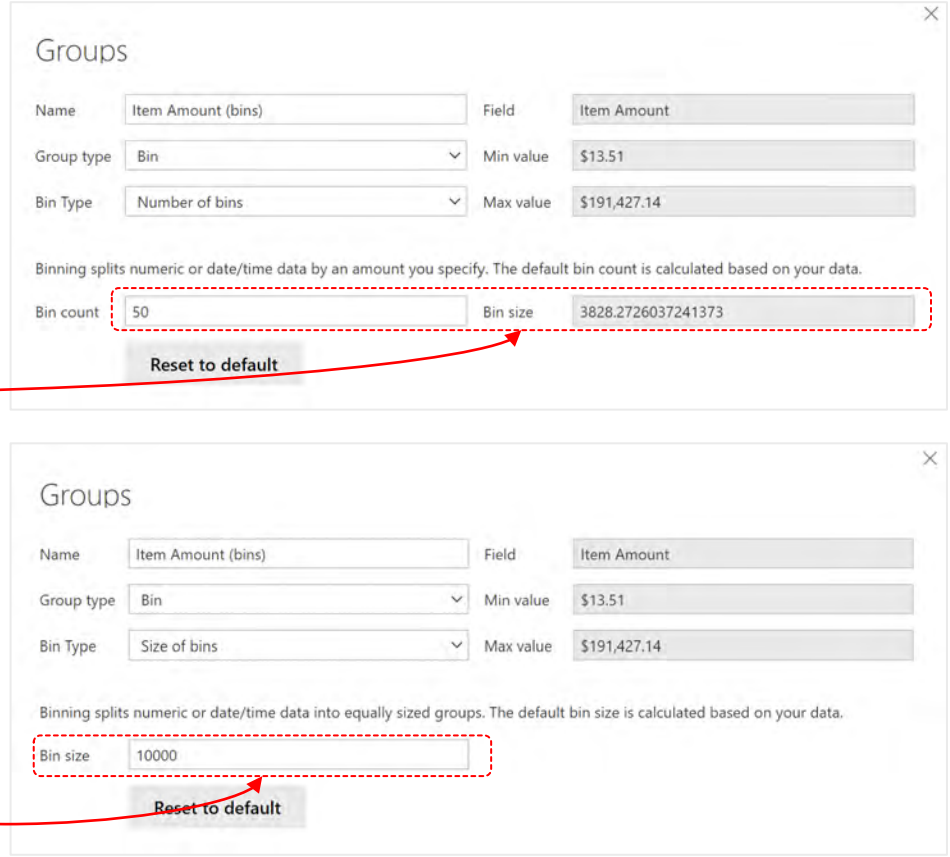


ORGANIZING DATA – GROUPS (NUMBERS)

"Size of bins" is usually the most useful. If we use "Number of Bins" and we have an odd Max value (as we do here) then setting an even number of bins gives an odd bin size.

In the example below setting a Bin count = 50 gives us a Bin size of 3828.27260.... which is not particularly useful.

However, setting a bin size to something sensible (e.g., $10,000) will give us something more useful to work with!

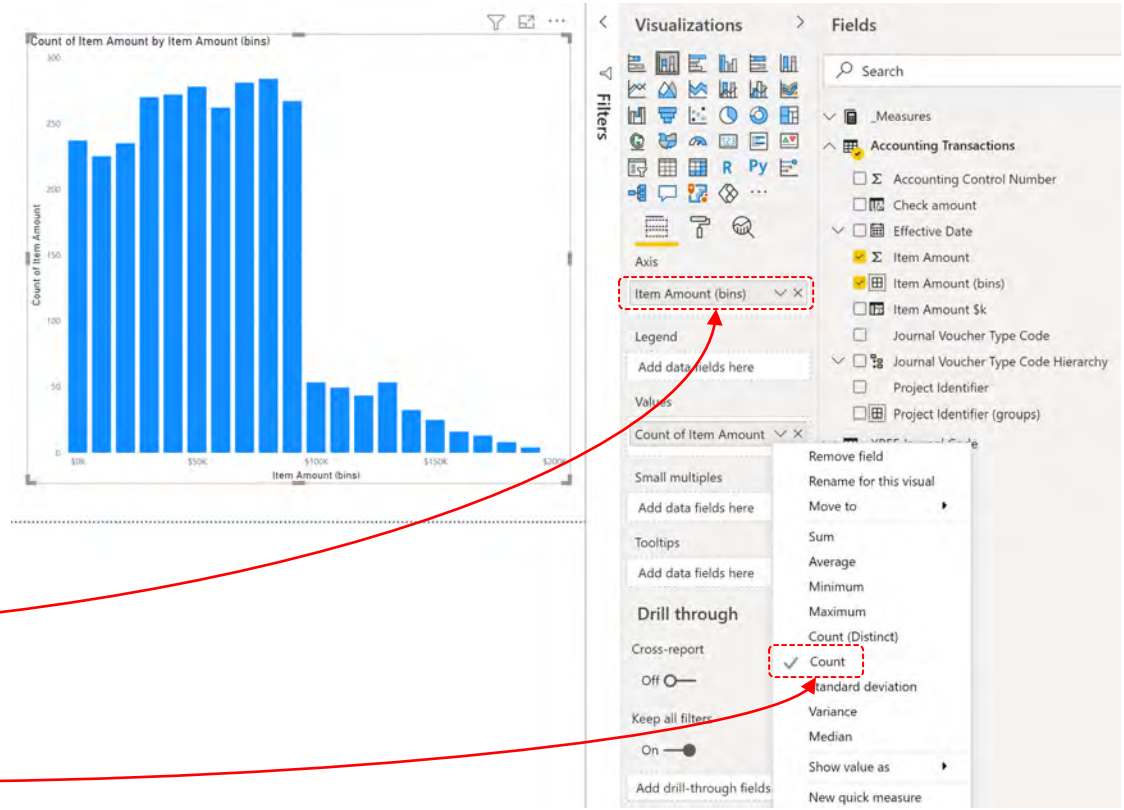


ORGANIZING DATA – GROUPS (NUMBERS)

Let's set the Bin size to $10,000. You will see a new column appear in the "Accounting Transactions" table called "Item Amount bins" (unless you changed the name in the Group dialog).

Let's use that new column to create a histogram.

1. In Report view add a Column Chart.
2. Drag the bins column to the axis.
3. Drag "Item Amount" to the values and in the dropdown select "count" rather than "sum".
4. Edit the bin size and see what happens!

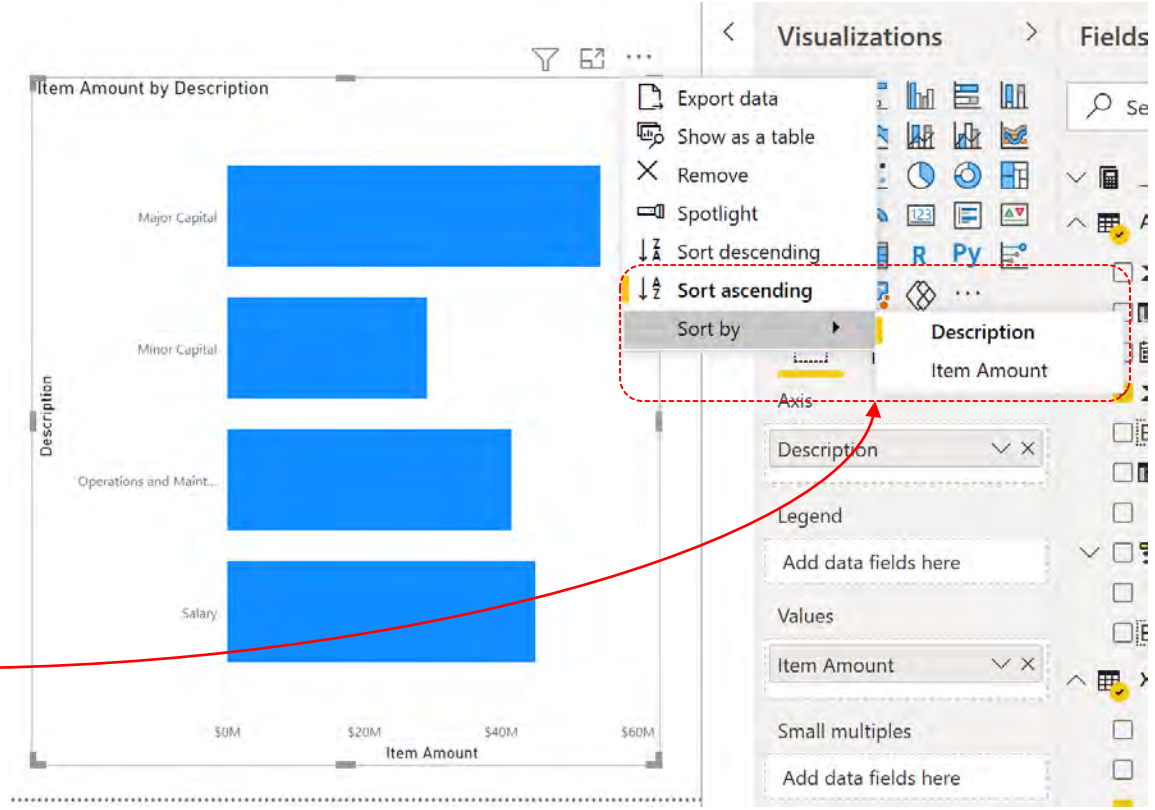


ORGANIZING DATA – CUSTOM SORT

If you click on a chart in Power BI, you can sort by the values and axis (ascending or descending).

You can also sort by another column. Follow the Instructor through the following steps

1. Create a bar chart with Description from "XREF Journal Code" as the axis and then "Item Amount" as the Values.
2. Click the 3 dots on the chart and select "Sort by Description" and check "Ascending".

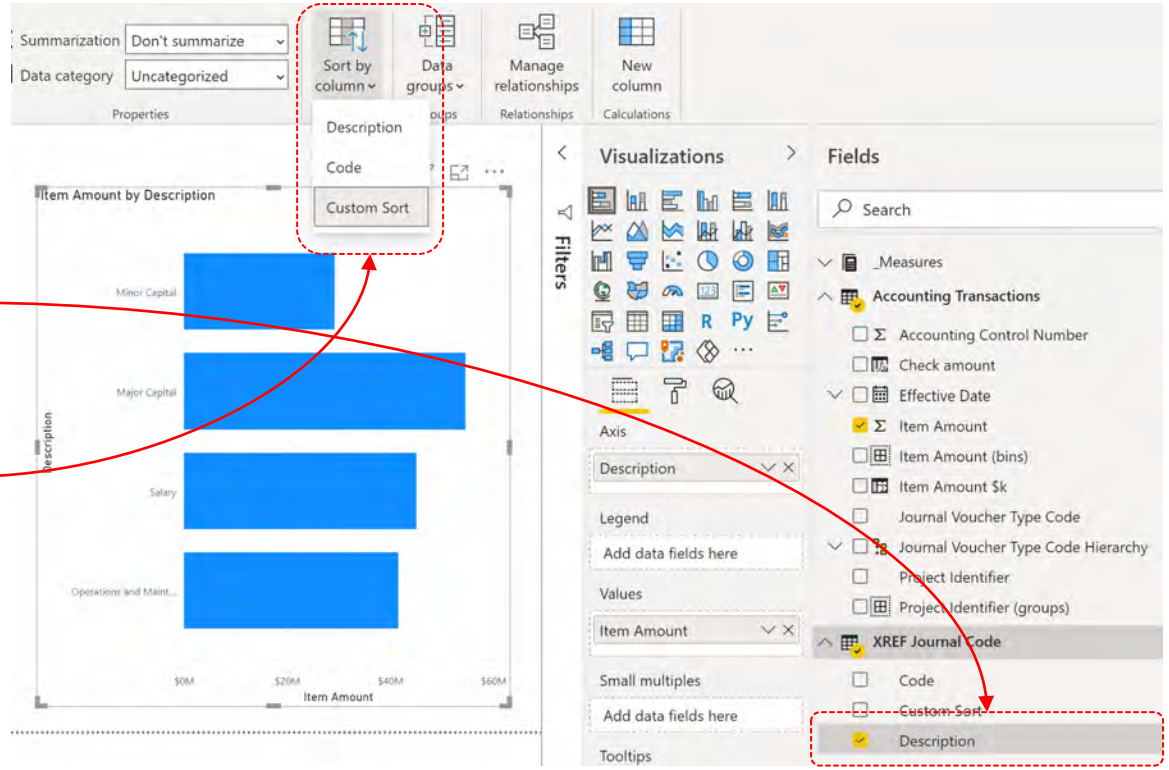


ORGANIZING DATA – CUSTOM SORT

If you look at the “XREF” Journal Code table, you will see a column in there called “Custom Sort”.

We want to sort by that column instead.

1. Click on the “Description” column name in the Fields tab.
2. In the Column Tools menu click on “Sort By Column and select “Custom Sort”.
3. If you want to change the sort order, go to the excel sheet and swap the numbers around then hit “Refresh”, the chart will update with the new sort order.



ORGANIZING DATA – CUSTOM SORT

There is another way to sort which is much more long winded, reasonably redundant but teaches us a new useful DAX command so we will do it!

Let us suppose that we want to create a chart just from one table (Accounting Transactions) and on the axis we are going to use “Journal Voucher Type Code”, but we still want to have the custom sort from the other table.

We need a way to grab the value from the other table and add it in as a column into “Accounting Transactions”, we can then use this to do our custom sort.

There are lots of reason to want to do this (e.g., to pull in a value from another table to do a row-by-row calculation) so even though it looks redundant learning how to do this is valuable!

ORGANIZING DATA – CUSTOM SORT

To achieve this, we are going to use the DAX RELATED operation. Follow the instructor through the following steps:

1. Click on the “Accounting Transactions” table and click on “New Column”.
2. Next type in the following:

```
Custom Sort Order = RELATED('XREF Journal Code'[Custom Sort])
```

3. The “RELATED” operation looks at all connected tables.
4. You then specify the column in that connected table you want to reference; one selected the code copies the data to the new column.
5. In essence it is very similar to using a “VLOOKUP” in Excel (although it isn’t really but the end result is similar.

ORGANIZING DATA – CUSTOM SORT

Now we have the custom sort in a new column do the following exercise:

1. Create a new bar chart.
2. Use “Journal Voucher Type Code” as the axis.
3. Use “Item Amount” as the Value.
4. Change the chart so that is sorts using the new “Custom Sort Order” column.

DATA WRANGLING

PBI-1: GETTING TO KNOW POWER BI

DATA WRANGLING

According to Wikipedia:

> “Data wrangling, sometimes referred to as data *munging*, is the process of transforming and mapping data from one "raw" data form into another format with the intent of making it more appropriate and valuable for a variety of downstream purposes such as analytics. The goal of data wrangling is to assure quality and useful data. Data analysts typically spend the majority of their time in the process of data wrangling compared to the actual analysis of the data.
>
> The process of data wrangling may include further munging, data visualization, data aggregation, training a statistical model, as well as many other potential uses. Data wrangling typically follows a set of general steps which begin with extracting the data in a raw form from the data source, "munging" the raw data (e.g., sorting) or parsing the data into predefined data structures, and finally depositing the resulting content into a, data sink for storage and future use.”

DATA WRANGLING

In this course we are going to be looking at a few, high level approaches to import and clean our data before we visualize it. We will do this all in Power Query, but it must be noted that some of these activities can be done in Power BI as well.

The reason to do the majority of Data Wrangling in Power Query is that only the post-processed data is loaded into Power BI, making the data set more streamlined.

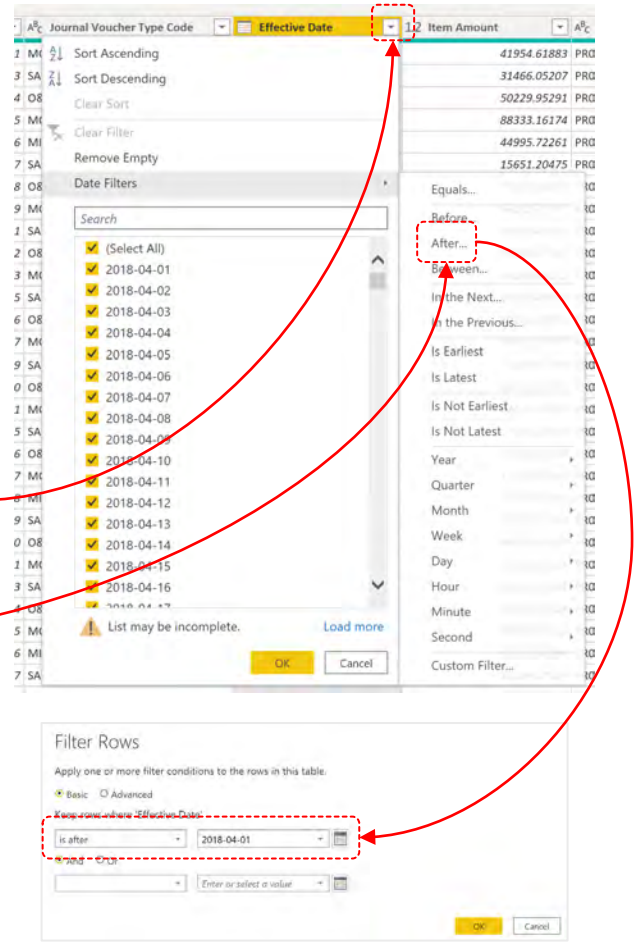
If we do our Data Wrangling in Power BI using DAX, then the entire data set has to be uploaded first which is typically less efficient.

In all of the following examples we need to get to our data in Power Query, to do that we first need to hit the “Transform Data” button in the “Home” menu.

DATA WRANGLING – REMOVING ROWS

The first way of removing rows in Power Query is by using the column filter. This is very similar to the "Filter" function in Excel.

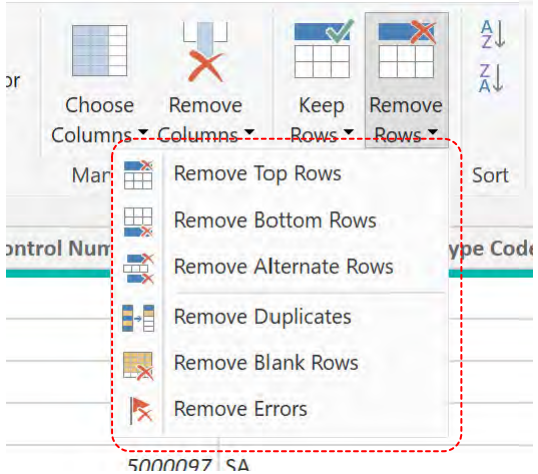
1. Go to the "Accounting Transactions" Query and click on the "Effective Date" column
2. We want to remove the dates before 1st April 2018. To do this click on the small down arrow
3. Click on Date Filters > After > Enter date > OK
4. Those rows are now removed completely from the data set and will NOT be loaded into Power BI



DATA WRANGLING – REMOVING ROWS

The second way to remove rows is the “Remove Rows” dialog in the “Home” menu

- “Remove Top Rows” enables you to remove the top “X” rows from a table. This is useful e.g., when people put in a header row in Excel
- “Remove Bottom Rows” is identical to the above except for rows at the end of a table
- “Remove Alternate Rows” allows you to remove alternate rows in a pattern, useful if your data contains rows and subtotals
- “Remove Duplicates” removes all duplicate rows
- “Remove Blank Rows” removes all blank rows
- “Remove Errors” removes rows with errors (careful, we do want to fix errors)

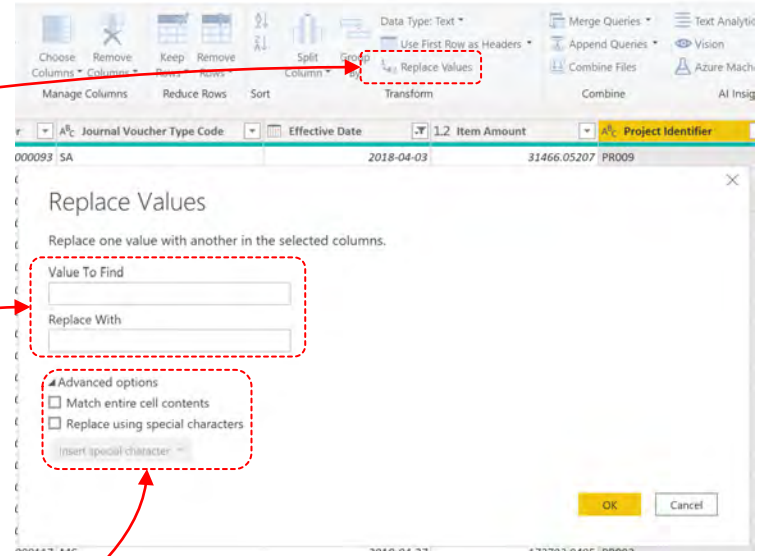


DATA WRANGLING – REPLACING VALUES

Replacing values is useful, e.g., we may have “N/A” in a row that is a date column.

Having the “N/A” means that we are unable to format that column as a date so in this case we would want to replace the “N/A” with a null (or empty) value.

1. Select a column you want to replace values on.
2. Click on the “Replace Values button which is in both the “Home” and “Transform” menus.
3. Type in the “Value To Find” and “Replace With”.
4. In the advanced options you get the option to match entire cell contents and to replace using special characters.

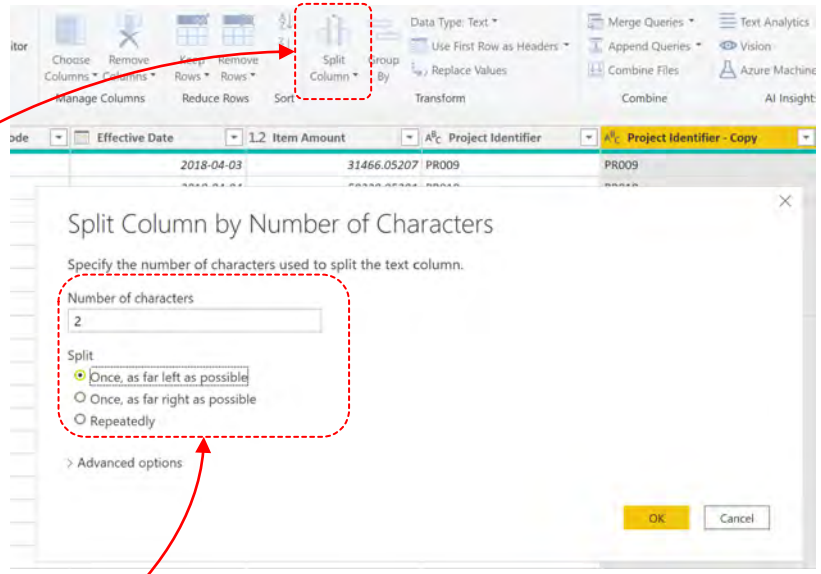


DATA WRANGLING – SPLITTING COLUMNS

Splitting a column into one (or more) separate columns can be a very useful thing to do.

For example, if you have a column with “Group – Level” (e.g., AS-01), it is advantageous to split these into two separate columns – “Group” and “Level” to create visualization.

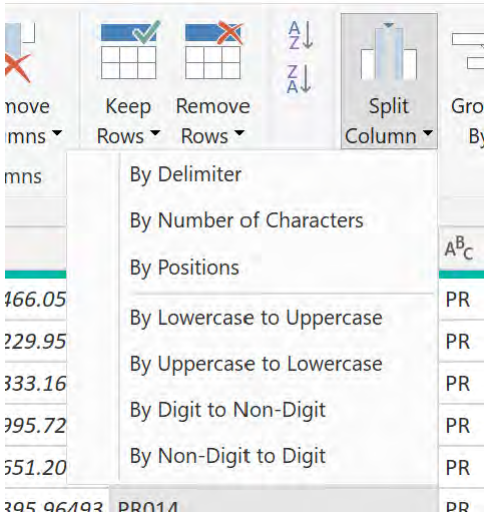
1. Select “Project Identifier” column from “Accounting Transactions”.
2. Right click on the column heading and select “Duplicate Column”.
3. Select the duplicate and click on “Split Column” button and select “by number of characters”.
4. Type in “2” and then select “Once as far left as possible”.
5. Rename the columns “Project Code” and “Project Number”.



DATA WRANGLING – SPLITTING COLUMNS

There are a few different options when it comes to splitting columns:

- By Delimiter – for example split at every instance of a comma.
- By Number of Characters – see previous example.
- By Positions – at fixed points in the string.
- By Lowercase to Uppercase – e.g., abcDEF would split into abc and then DEF.
- By Uppercase to Lowercase – as above but in reverse.
- By Digit to Non-Digit – e.g., 123ABC would split into 123 and ABC.
- By Non-Digit to Digit – as above but in reverse.



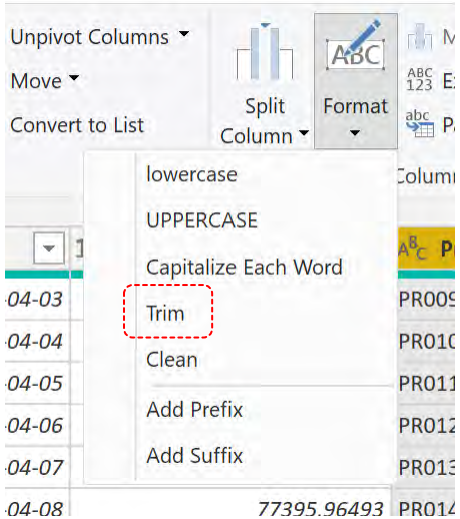
DATA WRANGLING – TRIMMING AND CLEANING

Often, we import data that has a space at the beginning or end of the record. These need to be removed, this is known as TRIMMING.

To do this select any column, select the “Format” button from the “Transform” menu and select “Trim”. All the leading and lagging spaces will be removed.

NOTE: that sometimes this does not work as there are two kinds of spaces, a regular space (ASCII code 32) and non-breaking space (ASCII code 160).

Non-breaking spaces will not be removed using Trim. Must use a special character search and replace instead (subject of a follow up course).

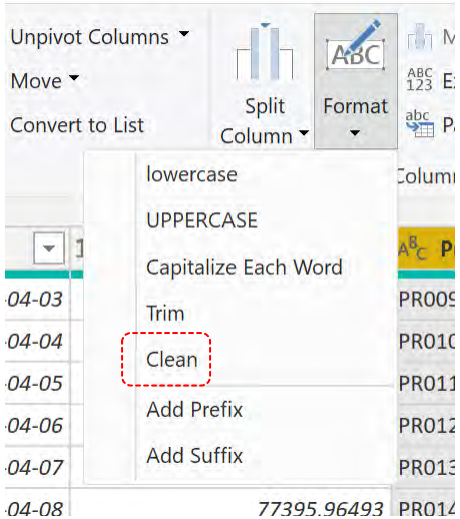


DATA WRANGLING – TRIMMING AND CLEANING

Sometimes our data contains unprintable (or non-printable) characters. These are things like carriage returns, tabs, line breaks etc. In fact, the first 32 codes, 0 – 31) are reserved as control codes for things like printers.

To remove these, perform the steps as per the previous slide but select "Clean" from the options.

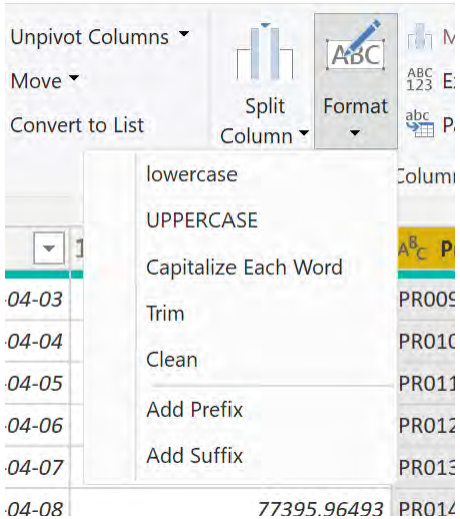
If your data contains a lot of text (e.g., comments or descriptions) it is best practice to perform a "Clean" step in your Power Query before importing the data.



DATA WRANGLING – OTHER FORMATTING OPTIONS

Other useful formatting options include:

- lowercase – change all characters in string to lowercase
- UPPERCASE – change all characters in string to UPPERCASE
- Capitalize Each Word – as described
- Trim – see previous example
- Clean – see previous example
- Add Prefix – add a text value to the front of each string in the column
- Add Suffix – add a text value to the end of each string in the column



DATA WRANGLING – APPENDING TABLES

Over the next few slides, we are going to work through an exercise that includes appending three tables together (we can do more if required). The exercise will also use several items we have also reviewed so is a good exercise to bring things together.

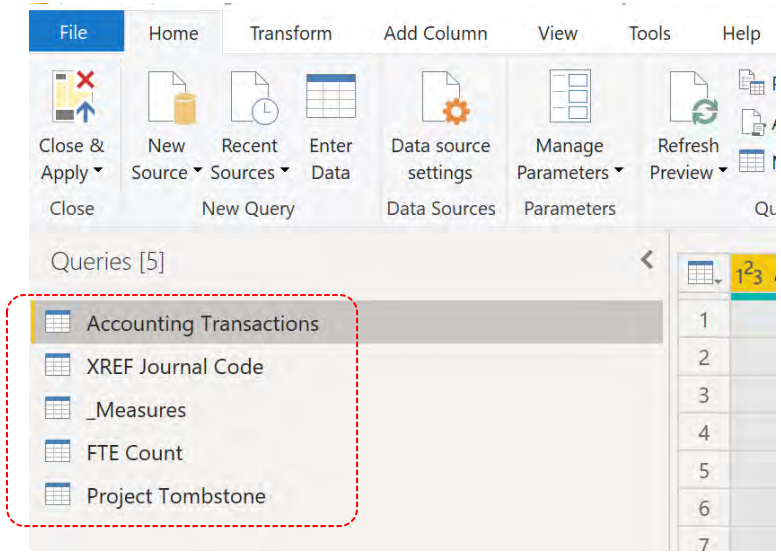
The problem we are going to solve is as follows:

1. We want to import new tables of data that contain “Project Identifier”.
2. We will create a cross reference table using “Project Identifier” to link all our tables together.
3. We COULD do this manually (a lot of people do) but what happens when new projects get added to either table?.
4. To solve this, we are going to dynamically create an automatic cross reference table from all of the “Project Identifier” columns.

DATA WRANGLING – APPENDING TABLES

The Instructor will guide you through the following steps:

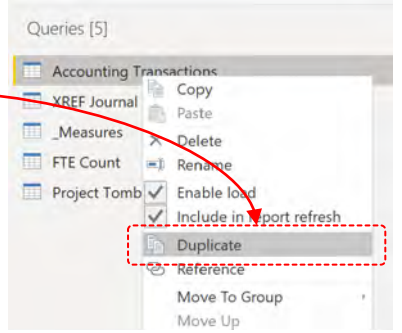
1. Import BOTH tabs from the “Data Set - Project.xlsx” (Project_Tombstone & FTE_Count).
2. Change the name of both the tables to remove the underscore (Optional).
3. Review all the columns in both tables to make sure that all the formatting is correct.



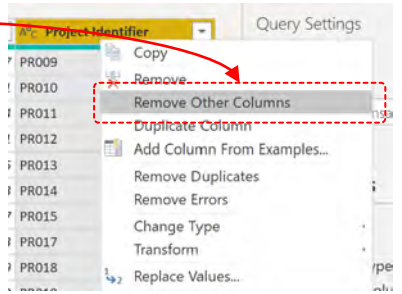
In the next part of the exercise, we are going to create a single XREF table that contains a set of unique “Project Identifiers” that we can use to link all of the tables together.

DATA WRANGLING – APPENDING TABLES

1. Select the “Accounting Transactions” Query, right click, select “Duplicate”.



2. Select the duplicate table and do the following:
    1. Rename it “XREF Project Identifier”.
    2. Select the “Project Identifier” column, right click and select “Remove Other Columns”.
    3. Do the same for “Project Tombstone” query but rename it “DNL Project Tombstone”.
    4. Do the same for “FTE Count” query but rename it “DNL FTE Count”.
    5. DNL stands for DO NOT LOAD – we will come back to that later.

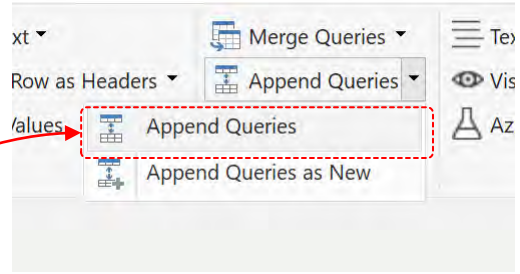


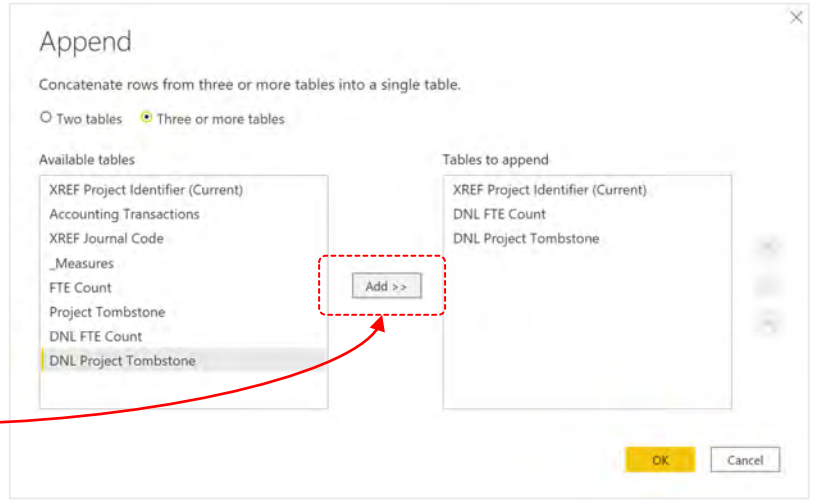
3. You will now have 3 tables each containing a single column which is “Project Identifier”. **NOTE:** the column names in each table need to be IDENTICAL!

DATA WRANGLING – APPENDING TABLES

Next, we are going to “Append” these tables, in other words stick them all together into a single table. To do that we need to click on our “target” table (the one which will be uploaded into Power BI). We can create a new table if we want but for now, we are going to use “XREF Project Identifier”

1. Click on “XREF Project Identifier”, then in the “Home” menu click on “Append Queries”. You will see 2 options, select “Append Queries” (not as new).
2. You will then see a new dialog asking you if you want to append two or three or more tables, select “Three or more”.
3. Click on “DNL FTE Count” then “ADD”, repeat for “DNL Project Tombstone”.

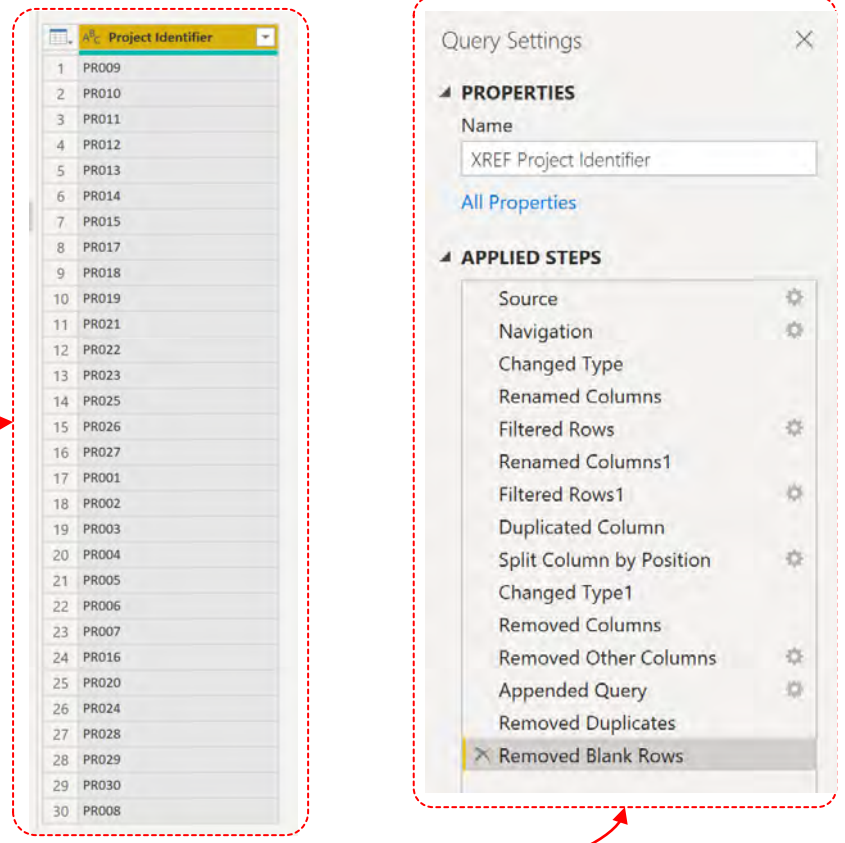




DATA WRANGLING – APPENDING TABLES

You might not see any different in the “XREF Project Identifier” query but the instructions now exist for Power Query to Append all the tables together. We now need to do a few things:

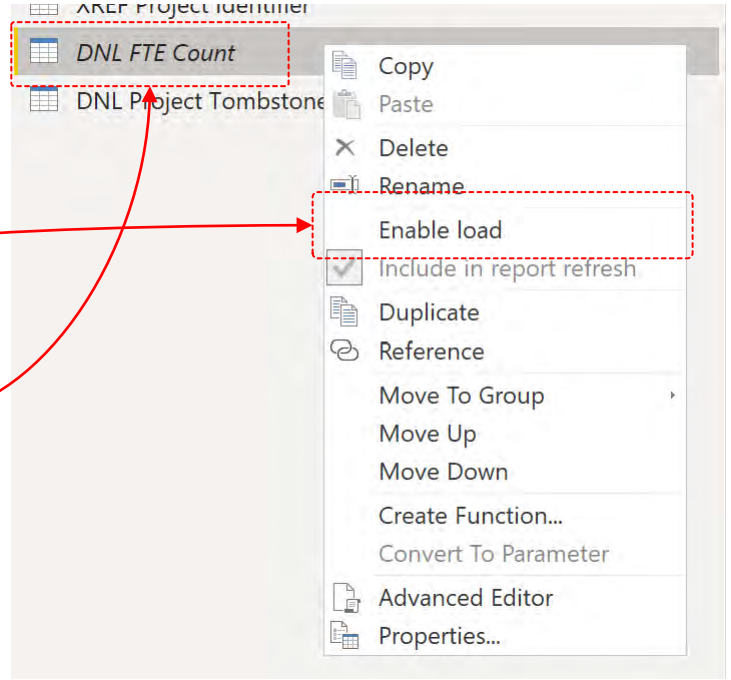
1. Apply TRIM to the “Project Identifier” column.
2. Apply Clean to the “Project Identifier” column.
3. Select “Remove Rows > Remove Duplicates”.
4. Select “Remove Rows > Remove Blank Rows”.
5. Your column will now contain a unique list of Project Identifiers based on ALL THREE TABLES.
6. See the image on the right for the query applied steps.



DATA WRANGLING – MANAGING DATA UPLOAD

We are still not quite finished. Remember those tables we renamed with the DNL prefix? If we just hit "Close and Apply" now, then they will be loaded into the Data Set and just confuse things, so we need to stop that:

1. Select "DNL FTE Count" query and right click. Then uncheck "Enable Load".
2. You know this has worked because the Query name changes to *italics.*
3. Repeat for the "DNL Project Tombstone" query.
4. Finally hit "Close and Apply" then once you are in Power BI hit Save.



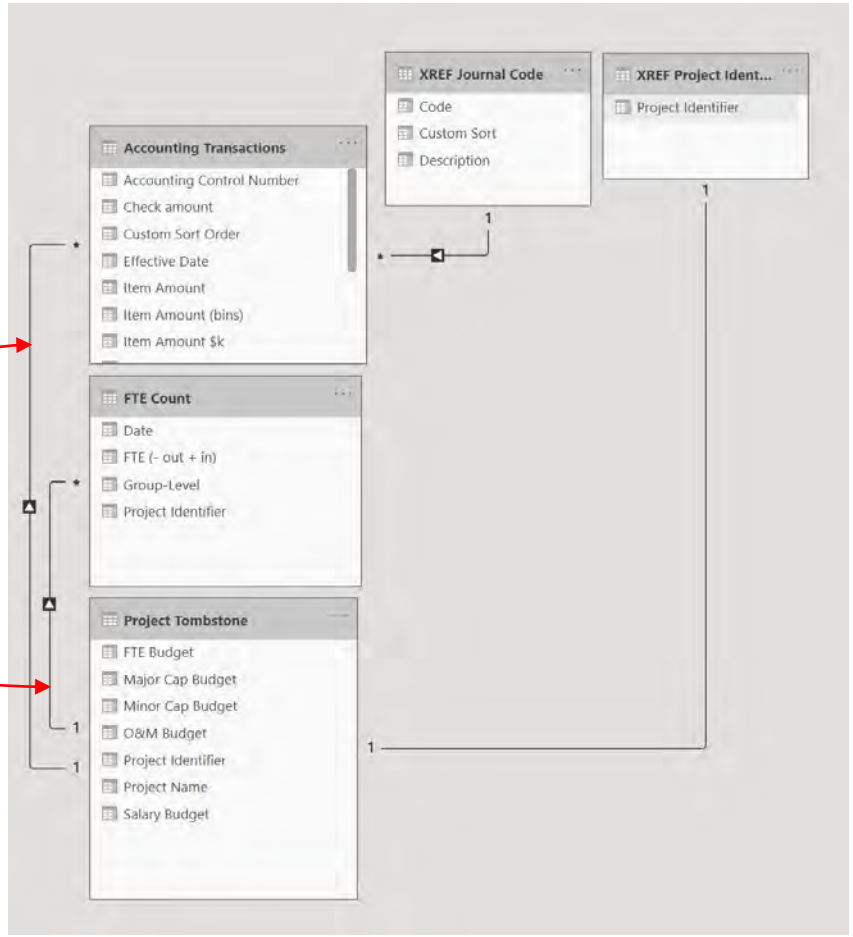
DATA WRANGLING – UPDATING THE DATA MODEL

We are now on our last phase. When we hit “Close and Apply” Power Query will try to link the tables together so let’s go to the modeling tab and look.

The initial view is very messy so move the tables closer together.

As we can see Power BI hasn’t got it correct so let’s delete the table connections that are not right.

Once we do that let’s create the correct connections.



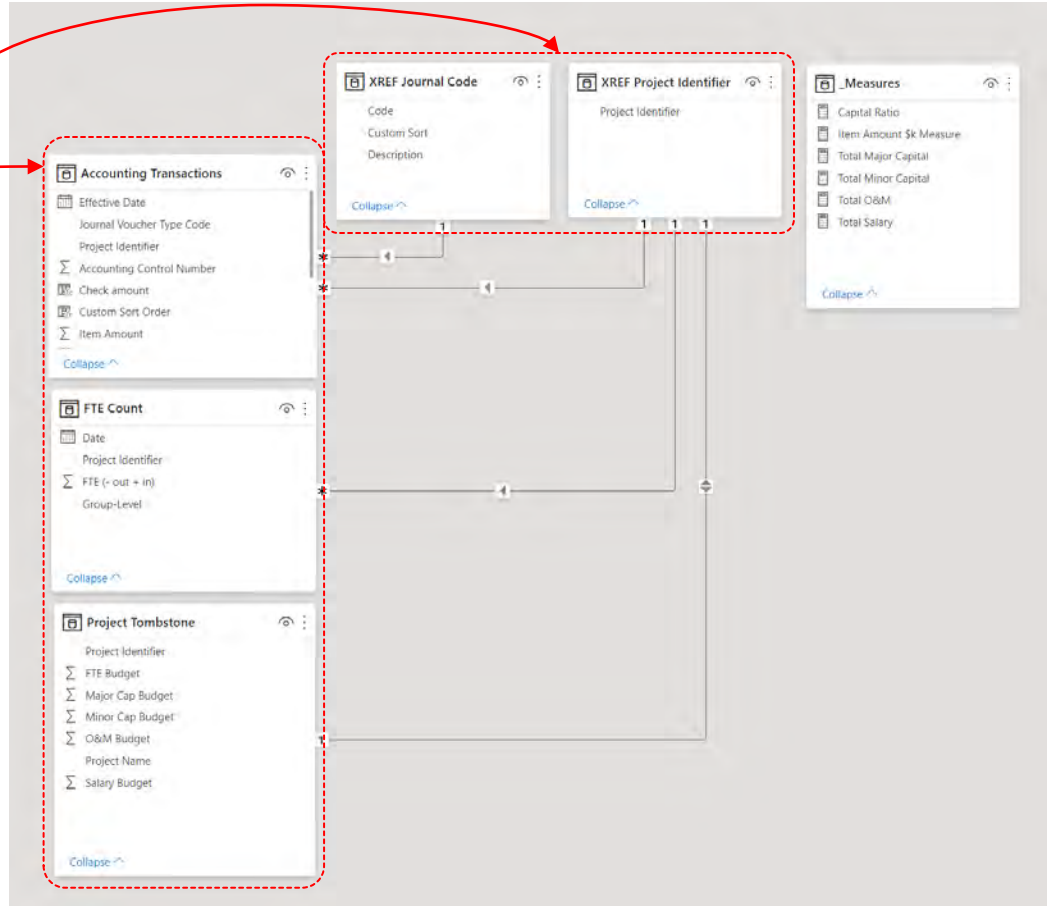
DATA WRANGLING – UPDATING THE DATA MODEL

It is good practice to have XREF tables across the top.

And to have data tables down the left.

This tends to make the data model easier to read.

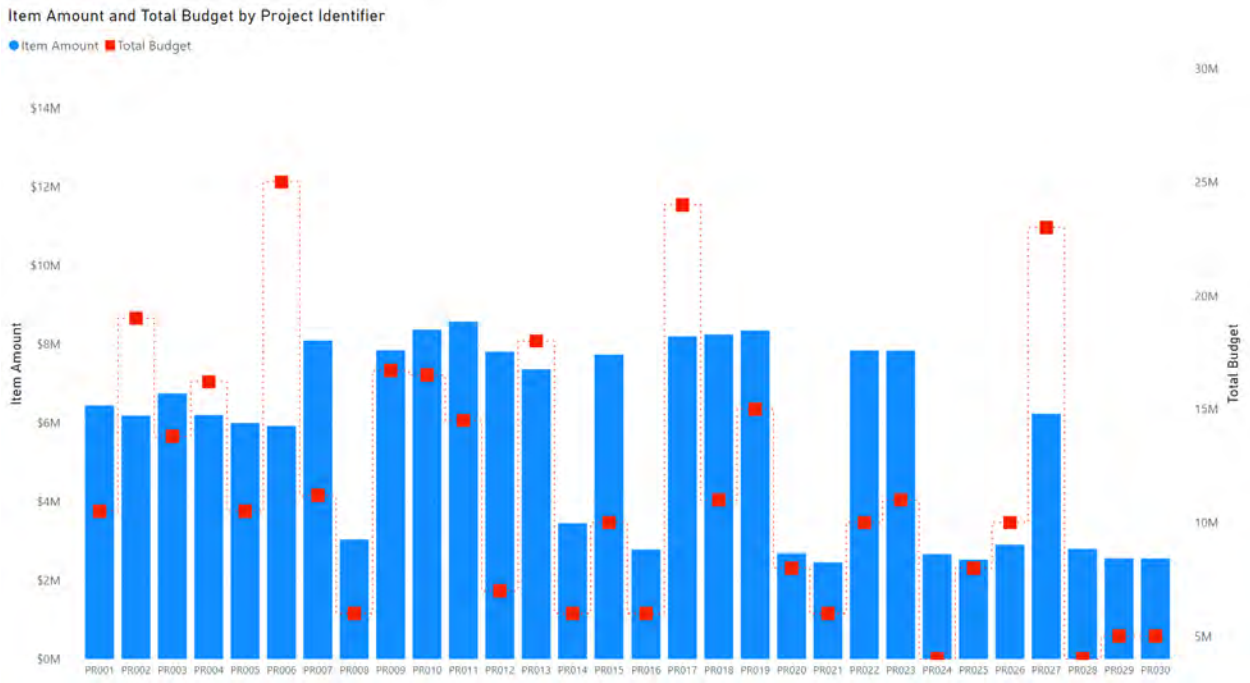
We can now create charts, measures, columns, slicers, etc. using “Project Identifier” from our new XREF table!



DATA WRANGLING – UPDATING THE DATA MODEL

Final exercise:

1. Create a chart that contains values from two or more of the tables with “Project Identifier” as the axis
2. Go into any one of the data files and add a new project on the bottom with a NEW Project Identifier that isn’t on the list
3. Save the excel and then update the Power BI and see what happens!



WRAP-UP

Remember:

1. Clean data is the only data
2. Let Power Query work for you, automate as much as you can
3. Create small, defined measures then look to combine them into larger ones
4. Think about your data model, create XREF tables to make visualizations work for you
5. Try out all the charts and see what works (yes even Pie charts!!!!)
6. Document your code
7. Follow all of the best practices
8. EXPERIMENT and SHARE – get as much feedback as you can!!

Questions?