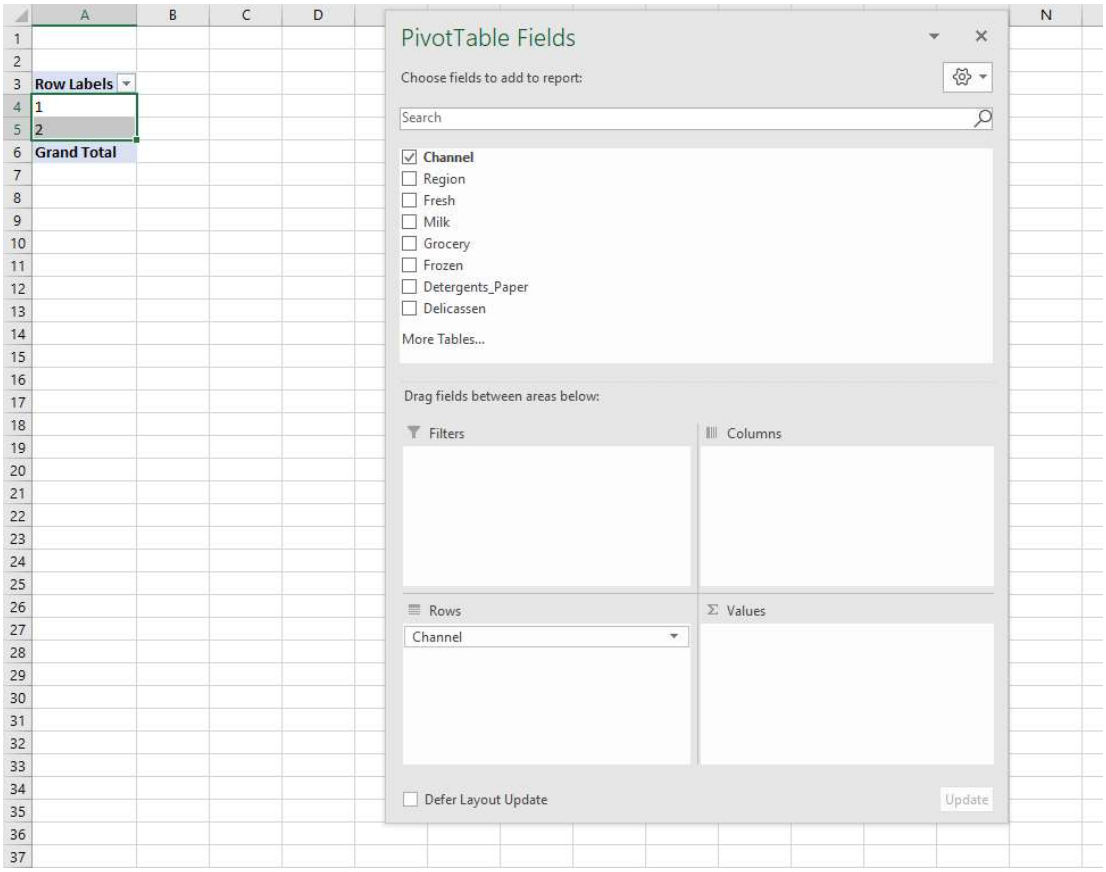
POWER QUERY AS EXCEL'S ETL – DEMO NOTES

- I want to "pivot" on the file `wholesale_customers.xlsx`. What is the problem? .
    - There is no "sales" field to "pivot" on. I have several fields representing one attribute, `category`.



- *How would we have fixed this in Excel without Power Query*
- *What does this example tell us about how data should be shaped?*

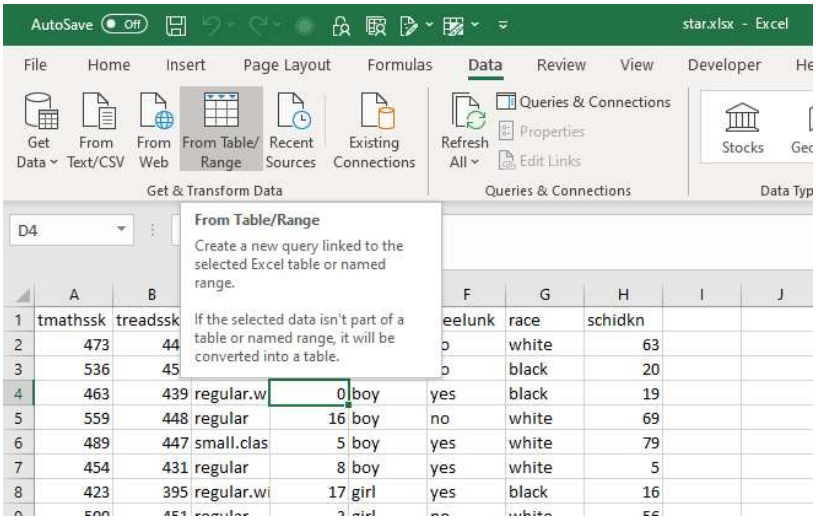
FIRST STEPS IN POWER QUERY – DEMO NOTES

Import a Table into Power Query

Demo: `star.xlsx`

1. Leave cursor anywhere inside the range you want to select

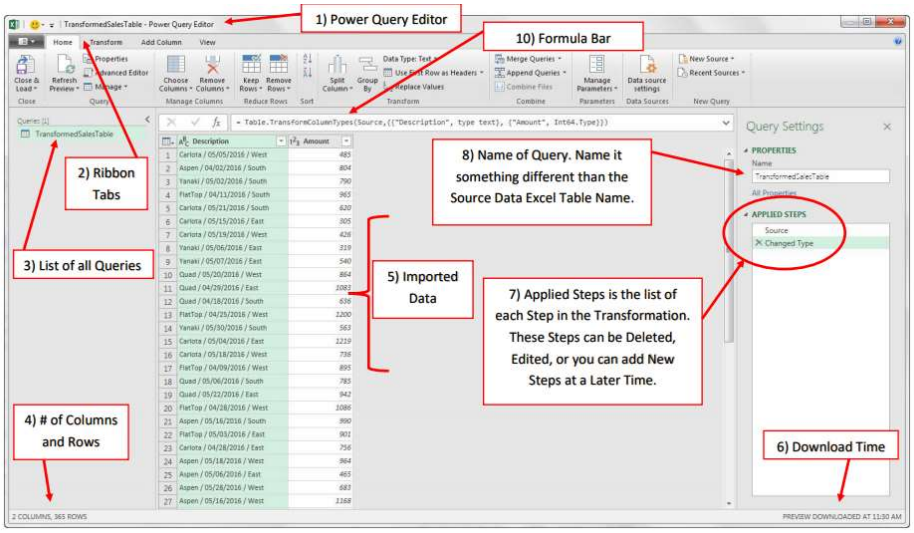
2. On the ribbon, select Data -> From Table/Range



	A	B	C	D	E	F	G	H	I	J
1	tmathssk	treadssk				eelunk	race	schidkn		
2	473	44				o	white	63		
3	536	45				o	black	20		
4	463	439	regular.w	0	boy	yes	black	19		
5	559	448	regular	16	boy	no	white	69		
6	489	447	small.clas	5	boy	yes	white	79		
7	454	431	regular	8	boy	yes	white	5		
8	423	395	regular.wi	17	girl	yes	black	16		

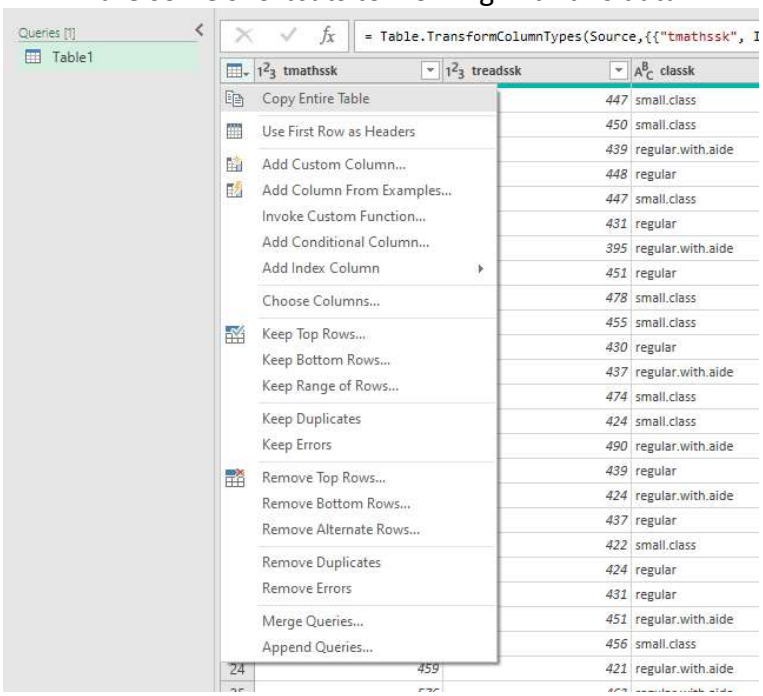
3. This will convert the range to a Table.

4. You will now see the Power Query Editor (source: https://people.highline.edu/mgirvin/AllClasses/348/MSPTDA/Content/PowerQuery/003-MSPTDA-IntroToPowerQuery.pdf)

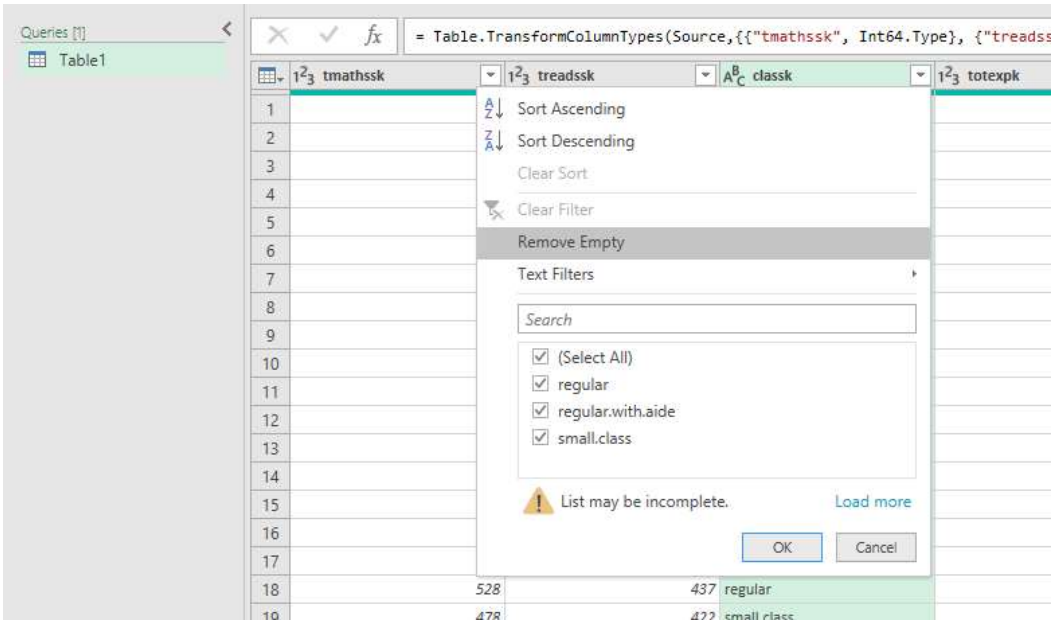


   a. A Home ribbon is at the top, just like in Excel. The first three tabs are going to have data cleaning functionality.

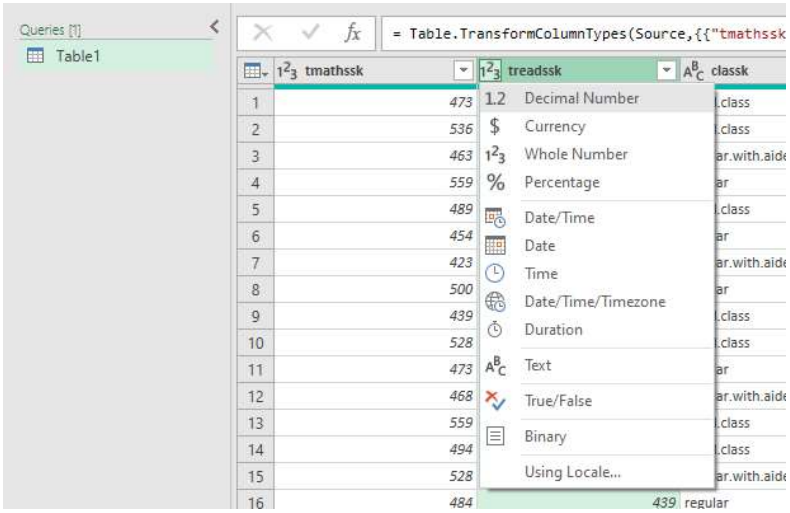
b. The imported data is in the middle of the screen. We can click on rows and cells and see their values at the bottom of the screen.
c. There is a small table icon in the “corner” of the dataset. Click on that and there are some shortcuts to working with this data.



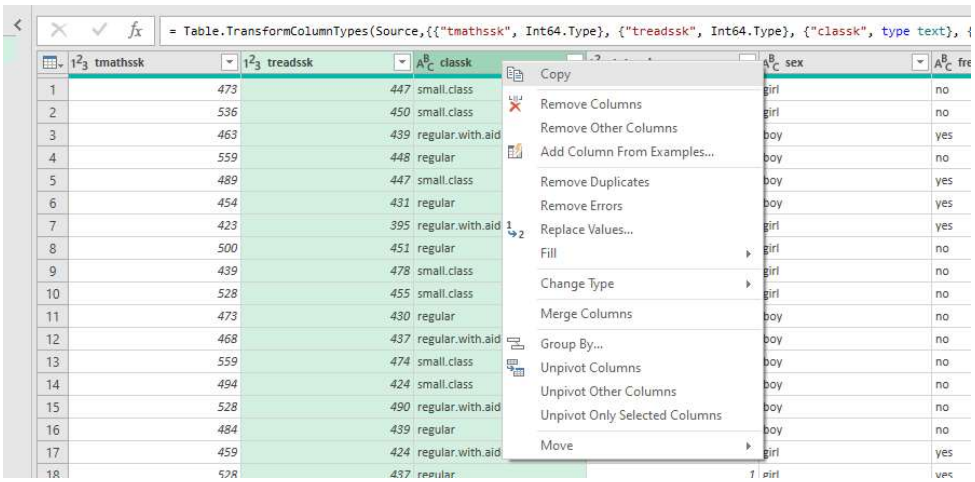
d. Click on any column drop-down and you’ll see you can filter it just like in native Excel.



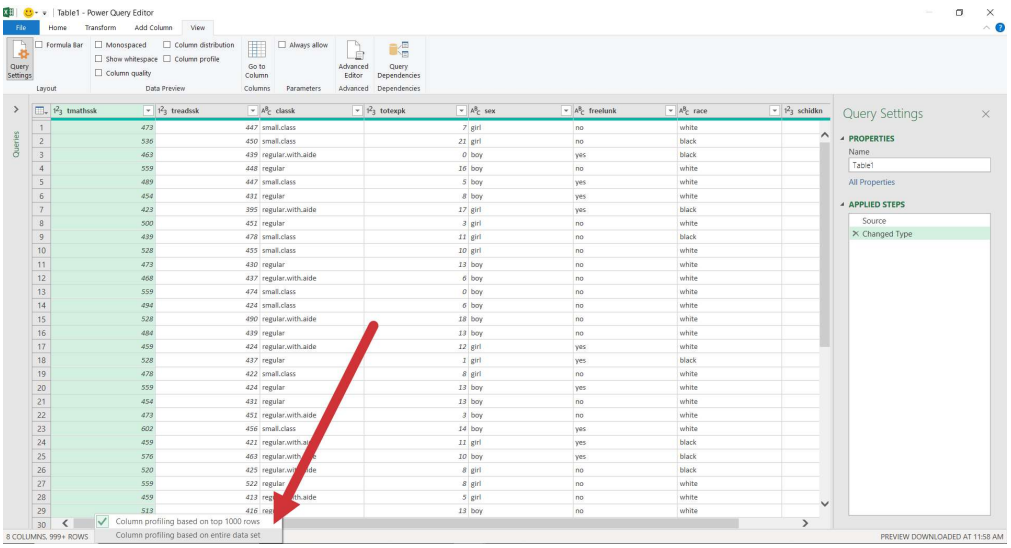
e. You'll also see a symbol to the left of the column. This indicates the column's type. You can click on that to change the data type.



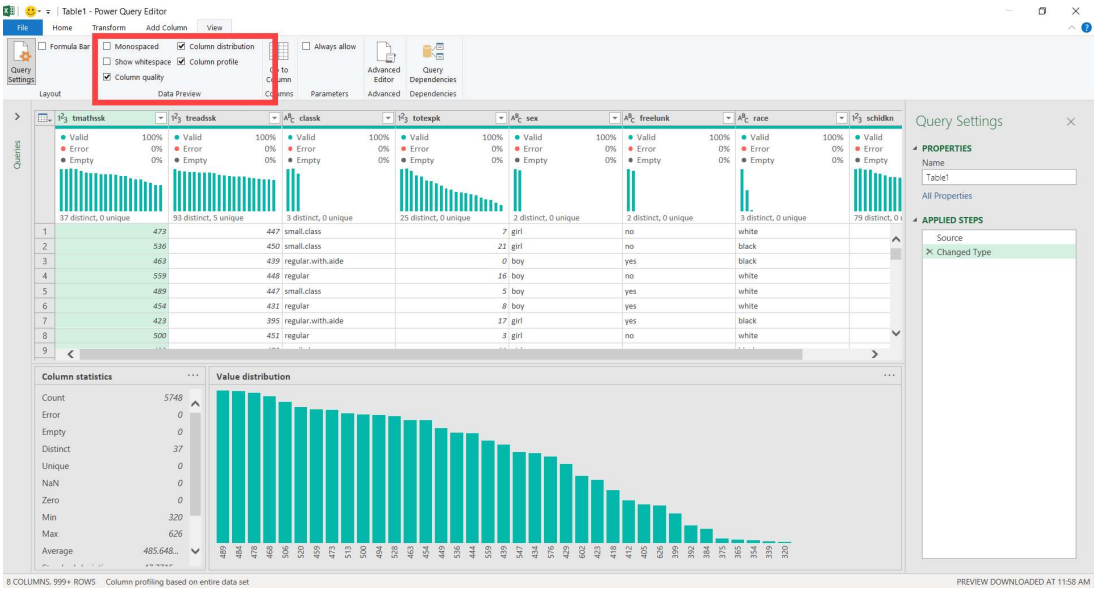
f. You can also right-click on a column to operate on it. Hold down Ctrl and click multiple columns to operate on multiple columns.



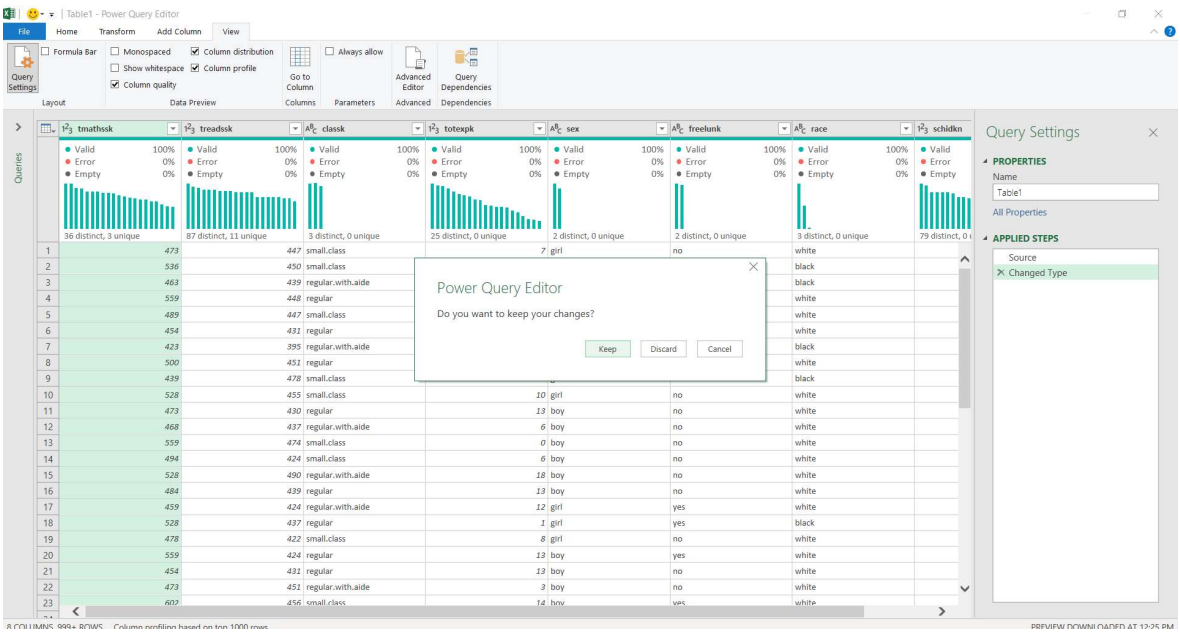
g. Now, go to the View tab on the home ribbon.
   i. Initially, what you are seeing in the Power Query editor is based on the first 1,000 rows.
   ii. To include all data in the Data Preview, click the message at the bottom which says `Column profiling based on top 1000 rows`. Change to `Column profiling based on entire data set`.



h. You can now change column appearance and add some statistics about each column using the Data Preview group of the View tab.



i. To exit the Power Query editor, hit the X on the upper-right. You can discard your changes for now.
    i. This will return you to “classic” Excel.



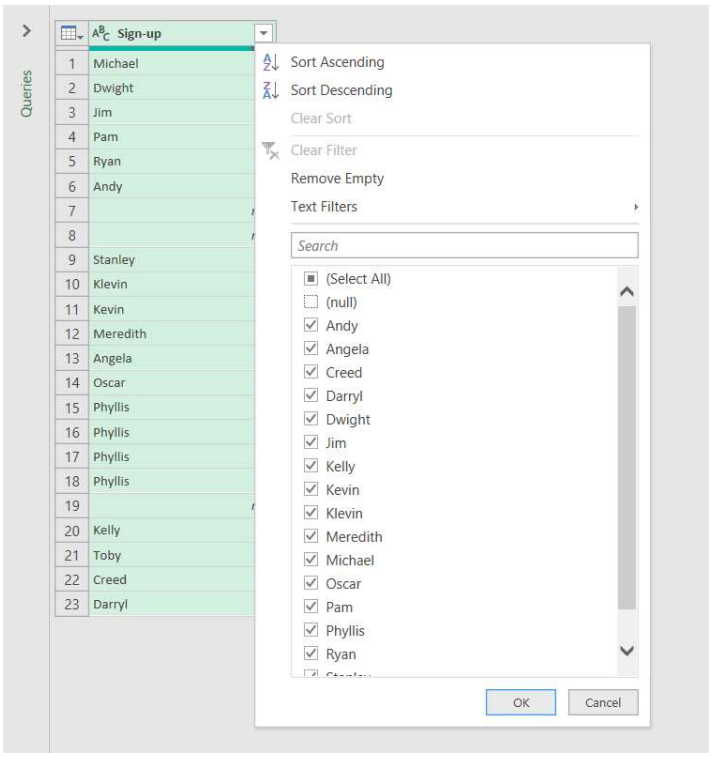
TRANSFORMING ROWS IN POWER QUERY – DEMO NOTES

Demo: `office-rsvps.xlsx`

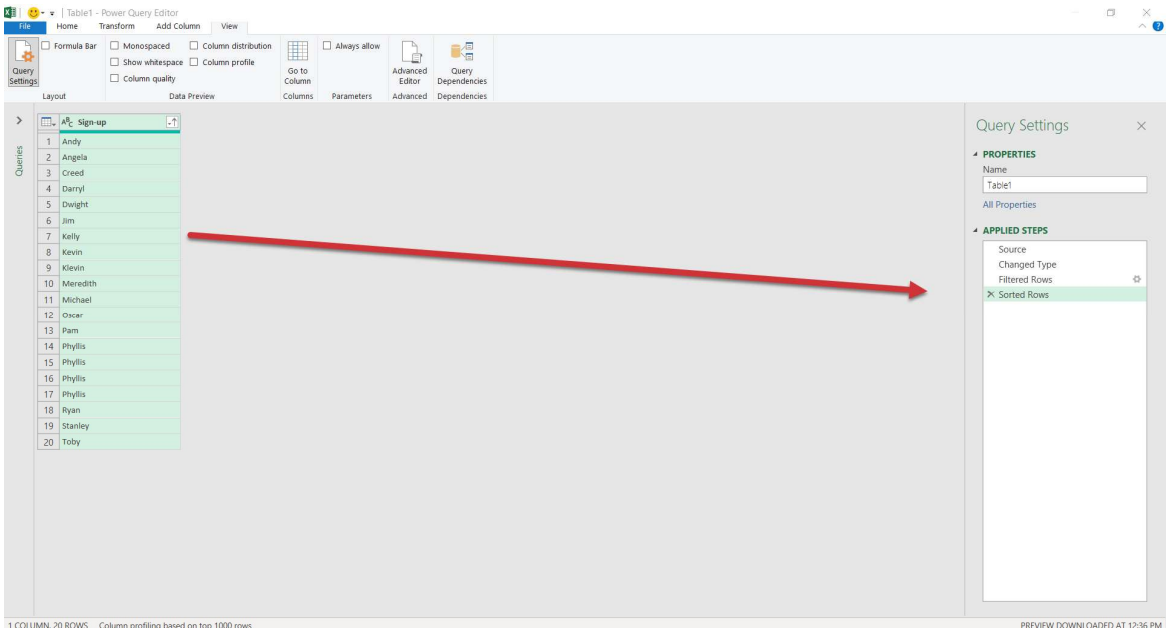
Worksheet: `signups`

1. You are consulting with the Party Planning Committee to clean up a list of RSVP's to a party. We would like to have the list sorted alphabetically, with duplicates, blanks and misprints removed.
2. This could be accomplished easily enough in classic Excel, but we would like to track each step of the data cleaning process, and we would like a solution that continues to work as more people RSVP to the list. These requirements make Power Query an excellent choice.
3. Create the connection from the range. Your data will be converted into a table.
4. You will see that blank values have been populated as `null` in Power Query. This is a special value indicating a missing value. It's not the same thing as zero!

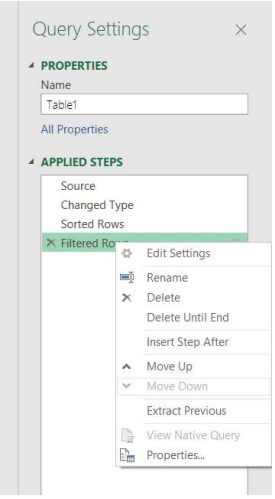
5. We want to filter out missing records, so select the drop-down on the column label and de-select `null`. This will remove them.



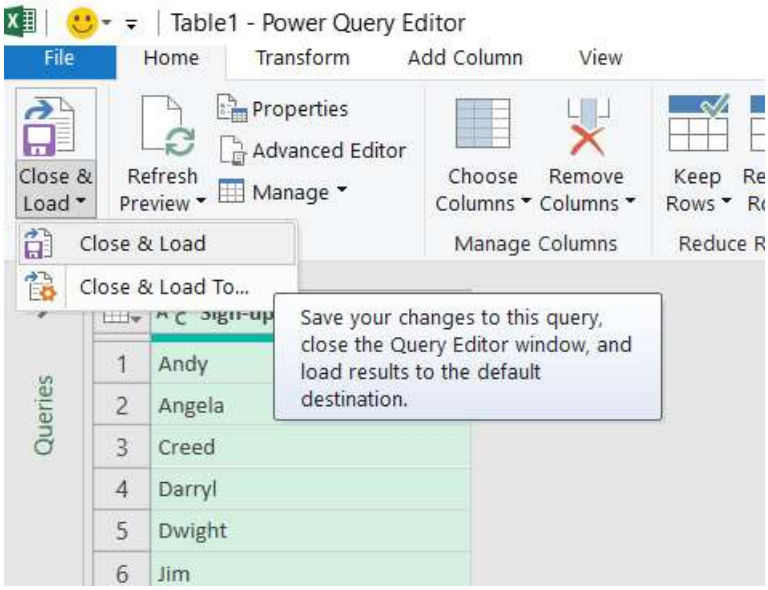
6. We can also sort the list A-Z with the same menu.
7. You will begin to see a running list of the steps we have taken on the right-hand side of the editor (Applied Steps).



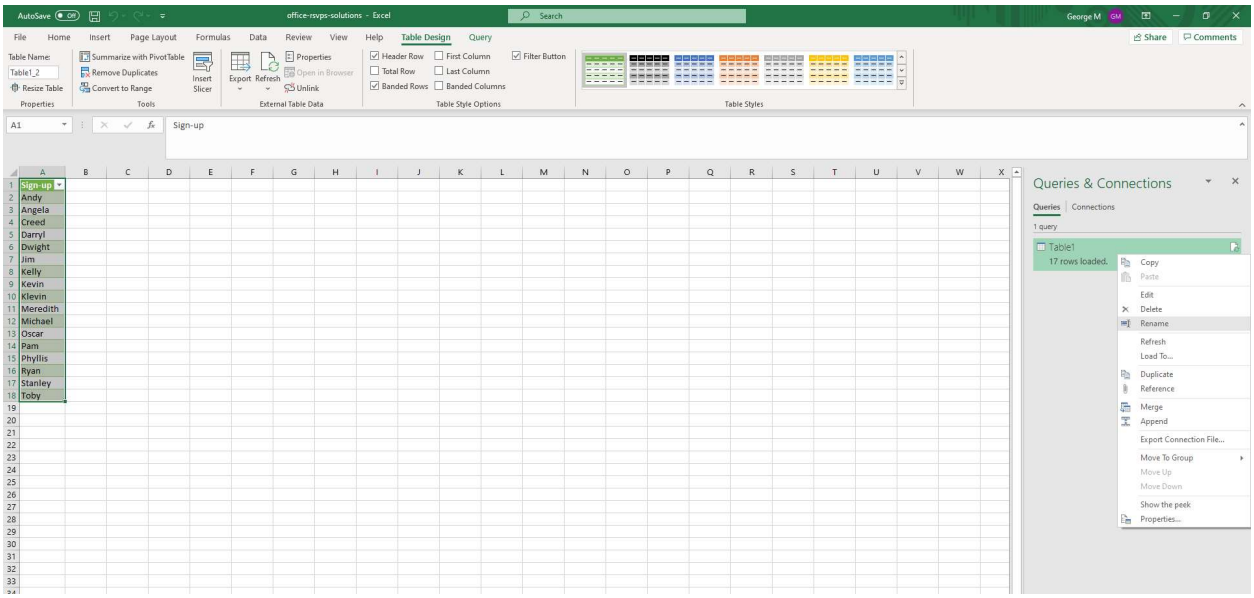
8. Let's remove the third step, Filtered Rows. Our dataset remains sorted A-Z, but nulls are no longer filtered out.
    a. **Careful: There is no "undo" for removing an applied step!**
9. Go ahead and re-filter the nulls from the data. You will see that becomes the last Applied Step.
10. You can modify the ordering of an Applied Step by right-clicking it.



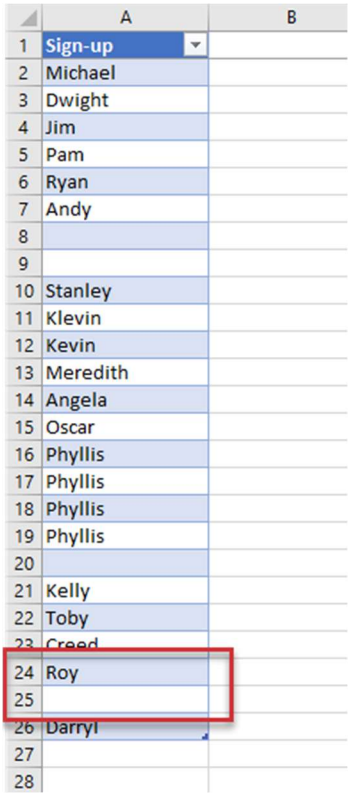
11. Remove duplicates by going to Home on the ribbon, then Remove Rows -> Remove Duplicates.
    a. You'll also see there is an option here to remove blank rows, this would have been another way to filter out nulls.
12. Last but not least, there is a misprint in the data: a `Klevin` in here. We don't want that either, so filter it out.
13. On the upper left of the Home tab, there is a Close & Load menu. Click the drop-down and select Close & Load.



14. The result of our query has been *loaded* back into Excel (the L part of ETL!).
15. To the right of our table is a Queries & Connections menu. Our query is named `Table1`. That's not a very descriptive name, so let's rename it to `party_rsvp`.
    a. If you want to close out this menu, you can open it again under Data -> Queries & Connections.

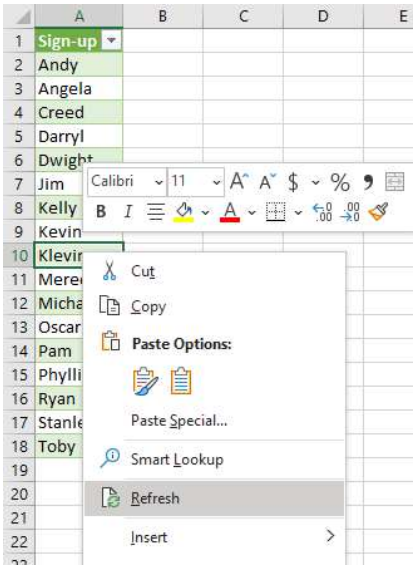


16. Now, any changes made to our source data will be re-loaded into Power Query, go through each step of the data-cleaning process, and be loaded into this new table upon refresh.
17. For an example, I am going to insert two lines into my table, Roy and a blank row.



	A
1	Sign-up
2	Michael
3	Dwight
4	Jim
5	Pam
6	Ryan
7	Andy
8	
9	
10	Stanley
11	Klevin
12	Kevin
13	Meredith
14	Angela
15	Oscar
16	Phyllis
17	Phyllis
18	Phyllis
19	Phyllis
20	
21	Kelly
22	Toby
23	Creed
24	Roy
25	
26	Darryl
27	
28	

18. Go back to the loaded query, right-click and select Refresh.

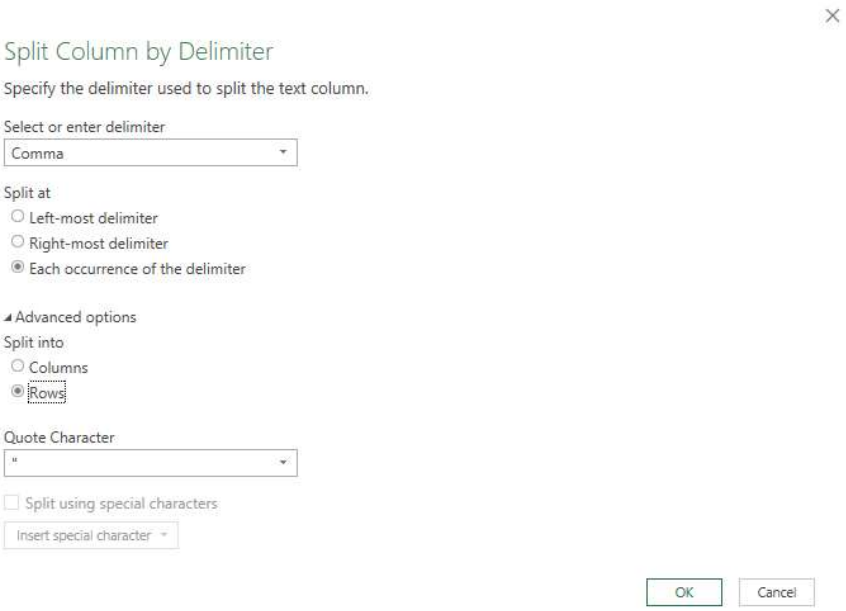


19. Roy made it into the RSVP, the blank didn't and the results remain sorted alphabetically!

Worksheet: `roster`

This time, the data has been created with commas separating each name by department. You would like to set up a report to automatically count how many people signed up from each department.

1. Bring the table into Power Query as usual.
2. Click the column and head to Transform > Split Column > By Delimiter.
3. We do want to split by each occurrence of a comma. We also want to click on Advanced Options and select "Split into Rows."

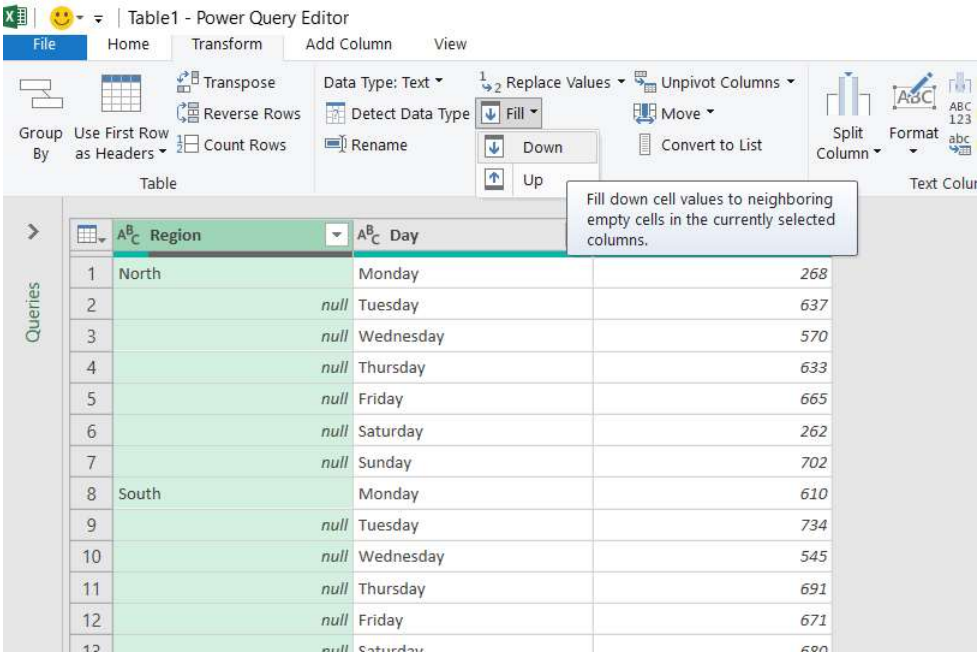


4. Click OK.
5. It looks like there is some leftover white space from this delimiting, so let's clean that up.
    a. Right-click on the column, select Transform and Trim.
6. Close and load. Now our data is tidy.

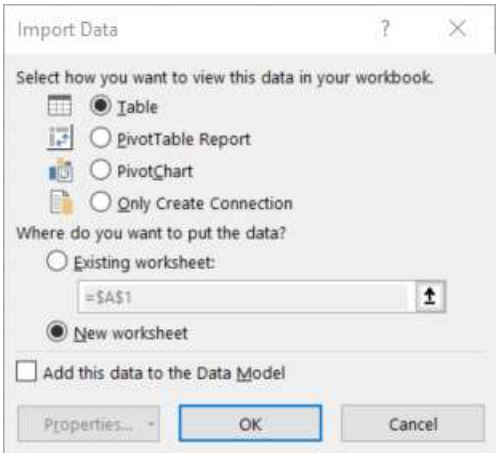
Demo: `regional-sales.xlsx`

1. This table does not have a header row and we need to fill down the Region fields. We would like to feed this data into a PivotTable for easy analysis.
2. Import our data into Power Query; remember that this time our Table does *not* have headers.
    a. We can rename the columns by double-clicking on them in the Query Editor. Name the three columns `Region`, `Day` and `Amount`, respectively.

3. To fill down the blanks for `Region`, highlight that column by clicking on it, then go to the Transform tab on the ribbon, you will select Fill, Fill Down to fill the nulls down with blanks.



4. We are ready to close and load this data. This time, select Close & Load To. This will give us some options for how to load the data:
    a. By default, Power Query loads into an Excel Table.
    b. We can also load it into a PivotTable or PivotChart. (PivotTable Report = PivotTable)
    c. Finally, there is the connection to only create connection. This means that the query is available in your workbook but not loaded into any worksheet.
    d. Note the checkmark at the bottom, “Add this data to the Data Model.” This would be if you wanted to build a relational schema in your workbook using Power Pivot.



5. Select PivotTable and we can build a PivotTable from the data just like any other dataset.

Drill: `state-populations.xlsx`

Worksheet: `states`

1. Name the query `State_populations`.
2. Remove the `United States` row from the data.
3. Fill down blanks on the `Region` and `Division` columns
4. Sort by `Population` from high to low
5. Load results into a PivotTable

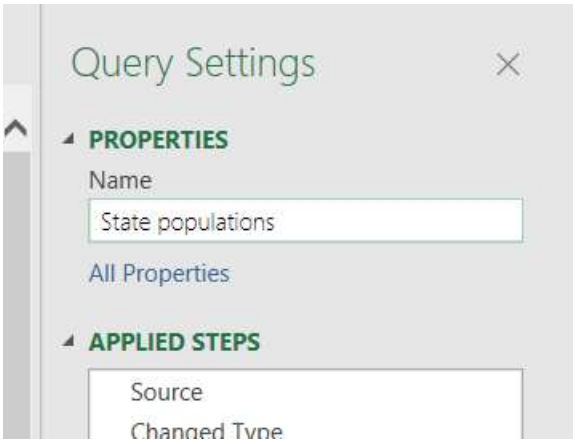
Worksheet: `midwest_cities`

These are the 50 largest cities in the Midwest.

1. Convert this data into a table so that each city is in its own row.

Notes for drill:

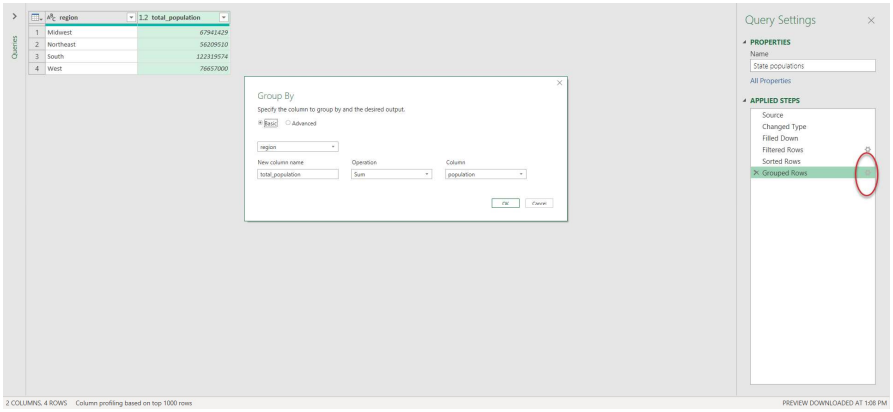
1. Operate on two columns at a time by holding down `Ctrl` and selecting each.
2. It's also possible to rename a query in the Query Settings menu within the Query Editor.



Additional demonstration on `State populations`:

1. It's possible to group/aggregate data in Power Query as you would using SUMIFS or a PivotTable.

2. As an example, right-click on `region` and select Group By.
3. For example, we can aggregate this data by total population by region by creating a new column `total_population` which is the sum of the `population` field.

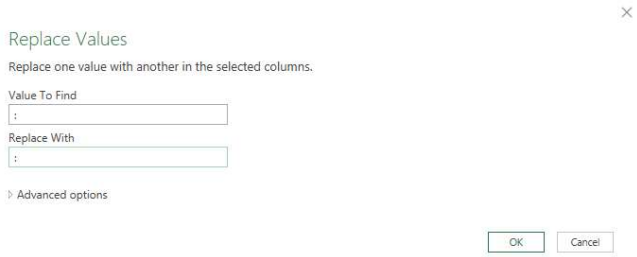


4. To look back at the settings of prior Applied Steps, click on the gear-wheel to the right of that step where applicable.

TRANSFORMING COLUMNS IN POWER QUERY I – DEMO NOTES

Demo: dvdrentals.xlsx

1. Create the query from the source table.
2. Convert Title and Artist Name to proper case by right-clicking the column and selecting Transform -> Capitalize Each Word.
3. There are no spaces after commas or colons. Add them by right-clicking on the headers and selecting Replace Values. Replace commas, then colons with each character followed by a space.



4. Split Item # into two columns based on the space delimiter by right-clicking on the column and selecting Split Column -> By Delimiter -> Space.
5. The UPC and ISBN 13 columns are probably better classified as strings than numbers. Change their types by clicking on the number icons to the left of their column headers and changing to text.

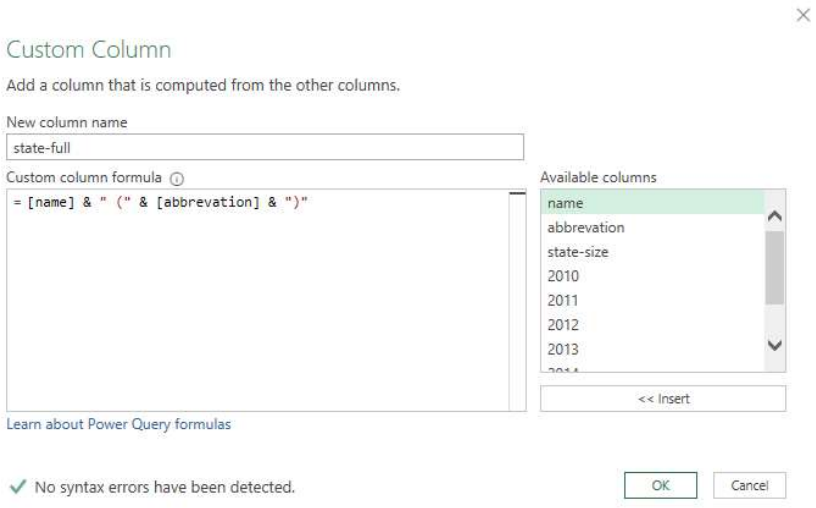
6. We don't need the BTKey column. Simply select it and hit Delete on your keyboard
7. Now, change the Retail column from Decimal to Currency.
8. Finally, convert the Release Date column into three columns, Year, Month and Day:
    a. Right-click the column label and select Duplicate column. Do this twice so there are three Release Date columns in total.
    b. Right-click the first one, and select Transform -> Year -> Year.
    c. Do the same for the remaining columns, but for Month and Day.
    d. Now, rename these columns as Year, Month and Day

Drill: orders.xlsx

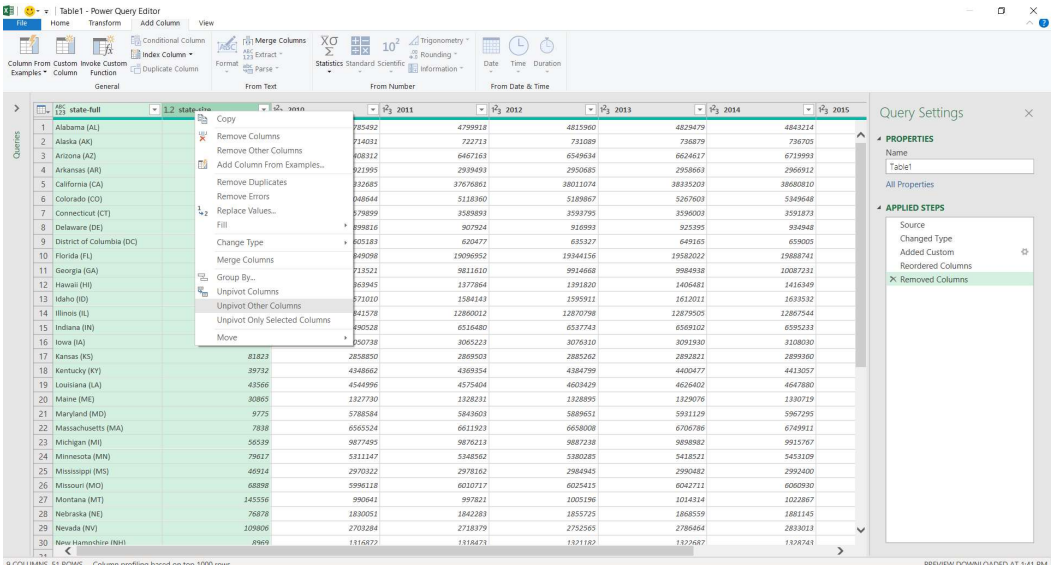
1. Convert the `Date` column to a month data type.
2. Convert the `Account` column to proper case.
3. Split the `Opportunity` column into three columns:
    A. `Vendor`
    B. `Status`
    C. `Order Type`

Demo: `population-densities.xlsx`

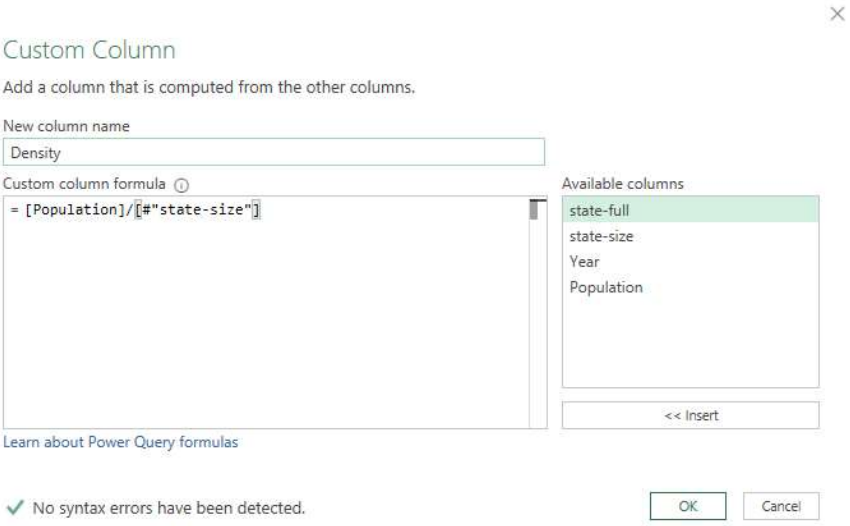
1. Load into Power Query
2. First, create a concatenated field in the format: `Name (Abbreviation)`
    a. Add Column on the ribbon, then Custom Column.
    b. Name this column `state-full`. Use ampersands to concatenate strings:



   c. Click link at the bottom of this menu to “Learn about Power Query formulas”: – this is M code.
   d. Our new column is added to Applied Steps. We can view the formula using the gear box.
3. Move the column to the front of this dataset by holding down Control and dragging it to the front.
4. Delete the two columns we had referred to in our formula. We can delete them and not break our calculated column.
5. We want to calculate population density. Rather than calculate the density for each year, we can “tidy” this dataset to get one, “population” variable, then calculate the density for each year in one fell swoop.
6. To create a “Year” column, select all but the `2010-2016` columns, then right-click and select Unpivot Other Columns.



7. We can re-name the `Attribute` and `Value` columns to Year and Population, respectively.
8. Now we can create another custom column formula, `Density`, which is `Population/state-size`.



9. Finished! Close & load.

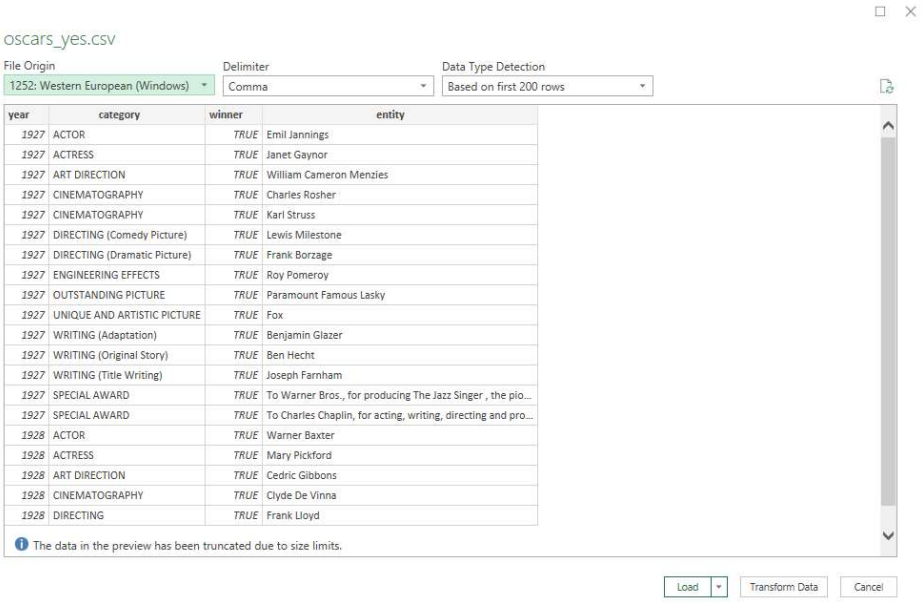
Drill: `wholesale-customers.xlsx`

1. Remember this data from the beginning of the lesson? Tidy it!
2. Create a field calculating 10% of the sales called `Tax`.

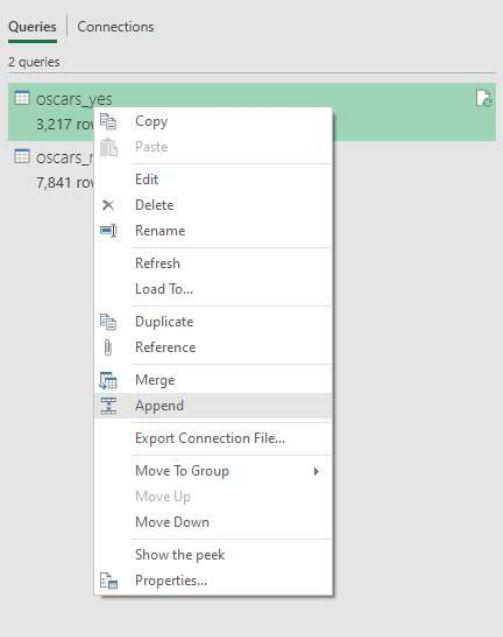
APPENDING TABLES IN POWER QUERY – DEMO NOTES

Demo: `oscars_yes.csv, oscars_no.csv`

1. Start with a blank workbook.
2. This time we will connect to a csv file. Still go to Data -> Get & Transform Data and select From Text/CSV.
    a. Connect to `oscars_yes.csv`
        i. Note: If we have the direct URL to the file, we can connect to the CSV file that way.
    b. An import menu will appear previewing the data. If we wanted to re-shape this data, we could select Transform Data at the bottom; however Excel seems to have done a good job with the import, so let's go ahead and load it to a table.

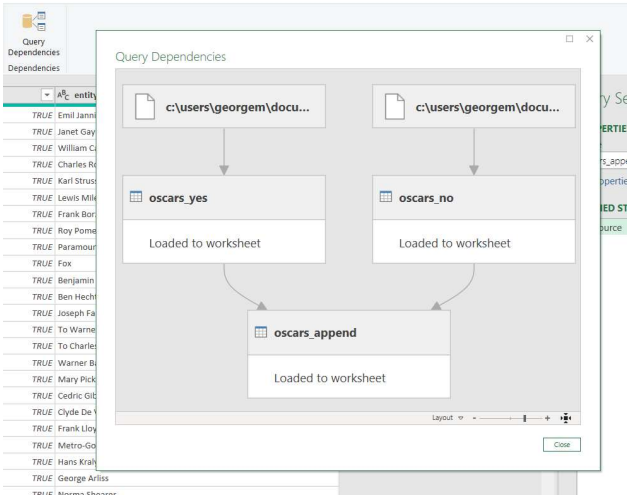


3. Do the same thing to export `oscars_no.csv` into this workbook.
4. There are now two queries in the Queries & Connections menu.
5. Right-click on the `oscars_yes` query and select Append.



6. Now we can append `oscars_no` to `oscars_yes`.

7. This will make a *new* query, named by default Append1. Rename it to `oscars_append`.
8. To get a visual look at how our workbook's queries are related, go to the View tab on the ribbon and select Query Dependencies.



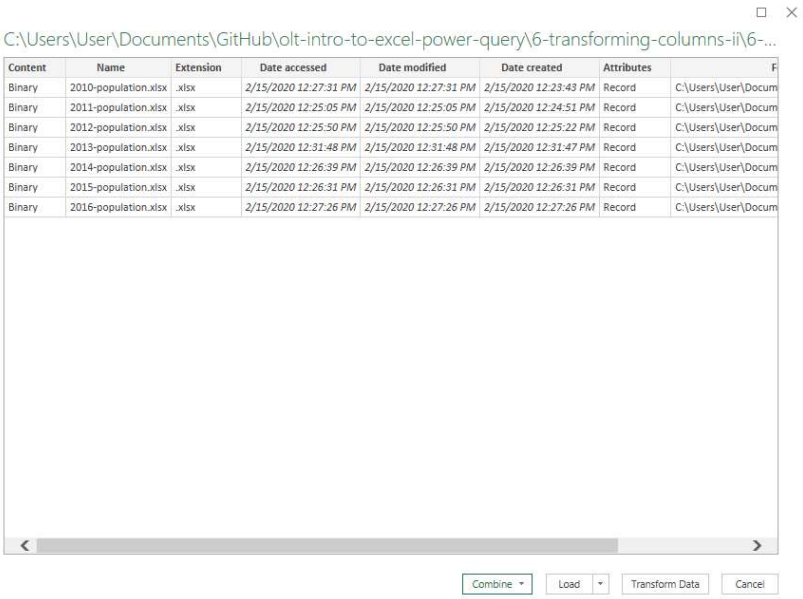
Drill: `hof_inducted.csv, hof_not_inducted.csv`

1. Append these tables.

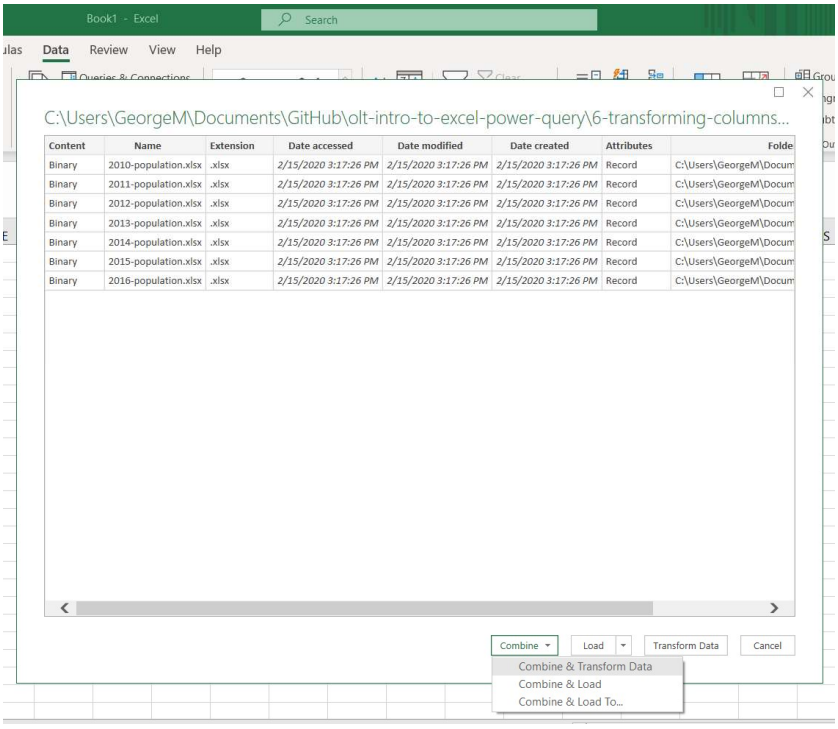
Demo: `state-populations folder`

This time we want to append the results of several files that are located in a folder. Instead of importing these in one at a time, we can read in the whole *folder* and append the data.

1. Open a blank Excel workbook and go to Data > Get Data > From File > From Folder
    a. Locate your state-populations folder. You are now going to see all of your files listed in this folder. That is pretty nifty already! We are going to take it a step further by appending these files together.
        i. To do that, select Combine > Combine & Transform Data

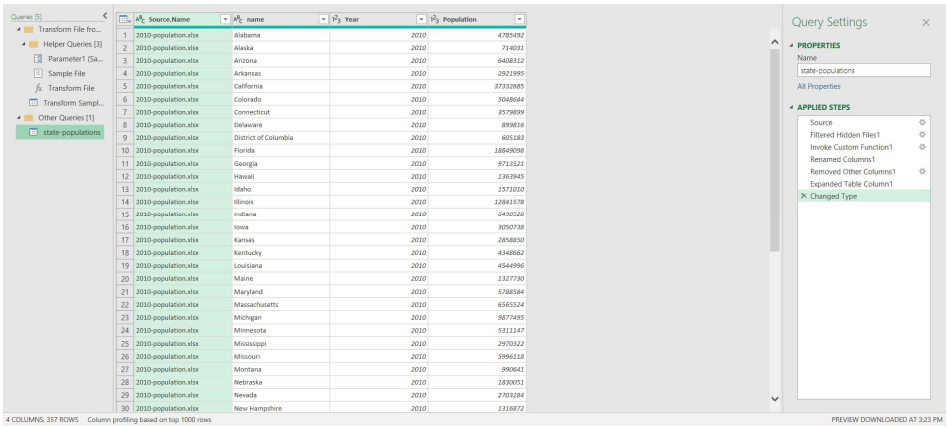


2. We now need to select what we should be extracting from each file. We only have one worksheet each named the same thing, so this is pretty easy. We will select “Combine & Transform Data.”



3. Click on the “state-population-worksheet” as the object that we want to extract from our files. This is the same across *all* files which will make this a lot easier for us.

4. Now you are going to see all these files have been appended together, we have a separate column for the file name, we can get rid of that if we want.
    a. Check out how we have a whole series of different queries to get to our result this time.



Drill: `baseball` folder

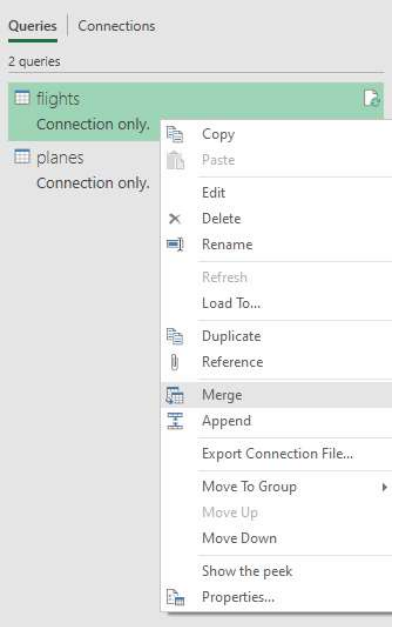
This is a download of the csv version of the [Lahman baseball database](#).

1. See if you can get a table of *all* files in this folder using Power Query.
    a. In this case we *do not* want to transform the data, just load a table of the file metadata.

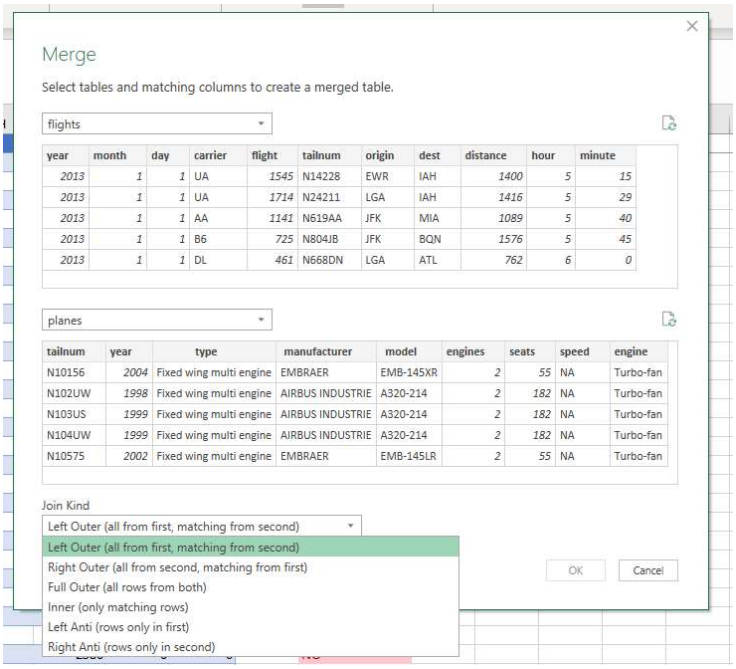
VLOOKUP(), MEET JOIN – DEMO NOTES

Demo: `flights-and-planes.xlsx`

1. We have a table of flights and tables of planes. The "lookup value" is `tailnum` but there is not a "match" for all of them (See `Found in planes?` column to confirm.)
    a. So, when we "look up" this plane information into our flights table, do we want to keep the information about the records without a match? Essentially we are asking, when we join `flights` on `planes` do we want to use a left outer join or an inner join?
2. Load both tables in Power Query and create only a connection for each.
3. In the Queries & Connections menu, right-click on flights and select Merge.

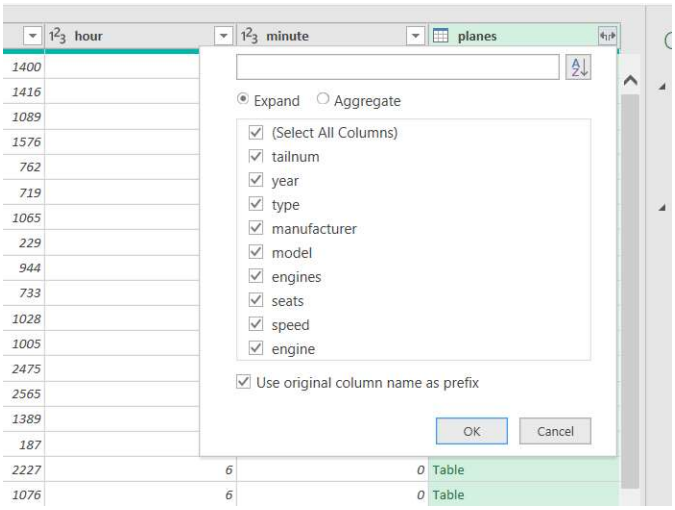


4. We will now create a merged table. We will merge flights on planes. Leave the Join Kind as Left Outer, but check out all the options available on the drop-down.

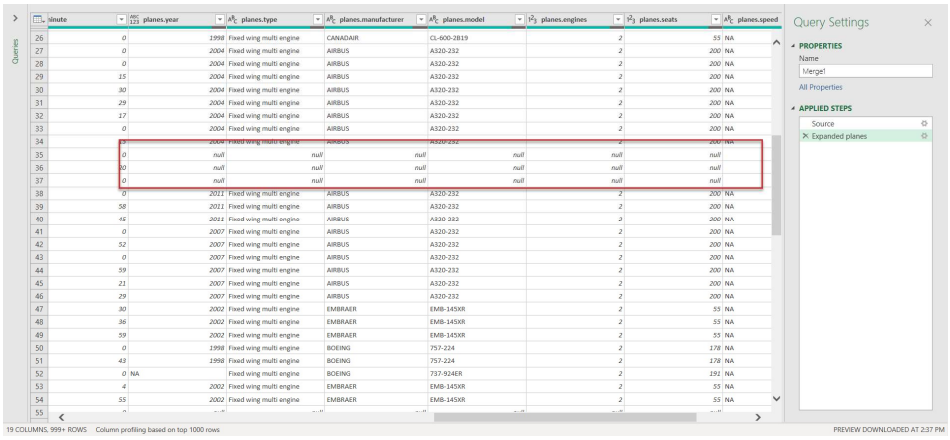


5. We can't hit OK until we specify *what* we want to join on. In VLOOKUP( )-ese, this would be our "lookup value" which in this case is `tailnum`.
6. We'll get a green check-mark saying it's matched X out of Y rows from the first table. We knew there were going to be some non-matches, so this number makes sense.

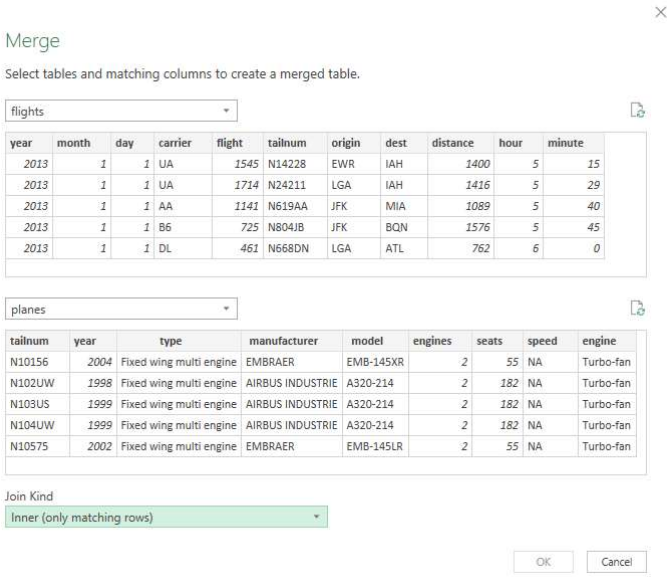
7. Hit OK, we get a new query, now we have an accordion-style menu here where we can select any of the returned fields into our merged table. We already have tailnum included in the table since that's what we joined on, so probably we don't need that one.



8. You'll see that each of these are named planes.field name. Undo our Expanded step to see why: Hit the accordion again. You'll see the option to "Use original column name as prefix" is checked on.
    a. This is not a terrible idea, for example there is a `year` field for the planes data and a `year` field for the flights field (one for when the plane was built, one for when the flight took place). So this way we easily know which is which.
9. Scroll down the resulting table and we can see there are rows of null's where there was no match for the planes data:



10. Now we can close and load the table and I am going to name it `left_join`.
11. Take the same steps except this time we will do an inner join of flights on planes.



12. Another green light.
13. Same steps, expand the resulting columns and load the table.
14. Check it out, this time there are only 284K rows loaded. Why? Well we can take a look here, there are no more NULL's for the plane info, those have been removed from the join. So it stands to reason there would be fewer rows this time.
15. Name the query `inner_join`.

Drill: `hof.csv, people-a-thru-m.csv`

1. What is the result of a left outer join of `hof` on `people-a-thru-m`?
2. What about an inner join?

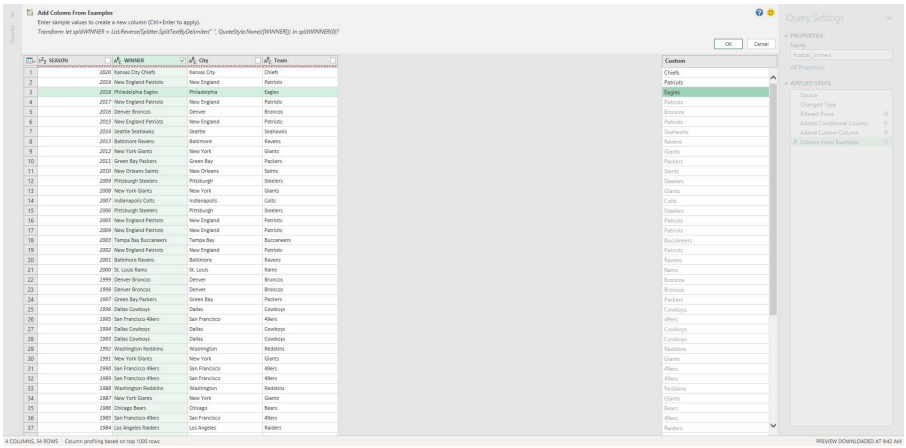
JOIN BEYOND THE BASICS – DEMO NOTES

Demo: `championships.xlsx`

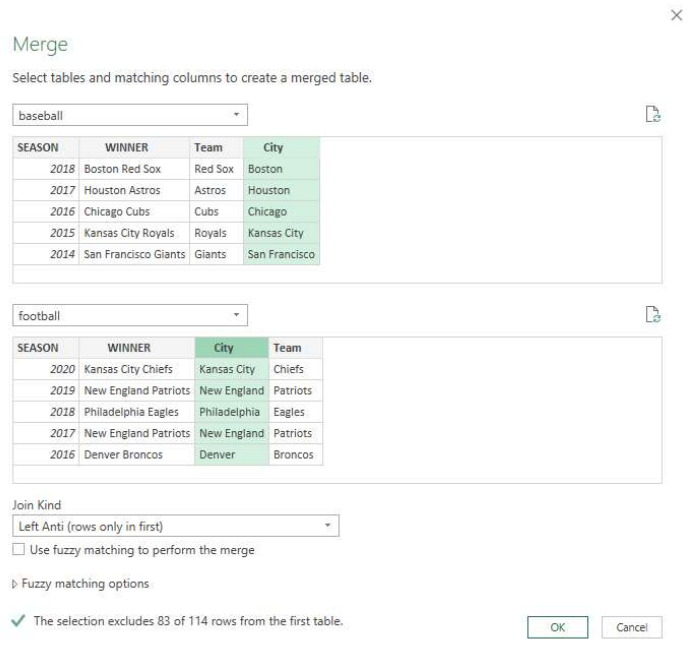
We would like to find what cities can claim *only* a baseball or football championship.

0. Preface: This data has been wrangled using Column From Examples. This is a powerful way to add a conditionally-formatted column to a table.
    a. To do this, open the football query, select WINNER field and head to the query editor and Add Column > Column From Examples > From Selection.
    b. What we want to do is start typing the name of the team in the new column. Power Query will start to use conditional logic to begin to complete the field for us.

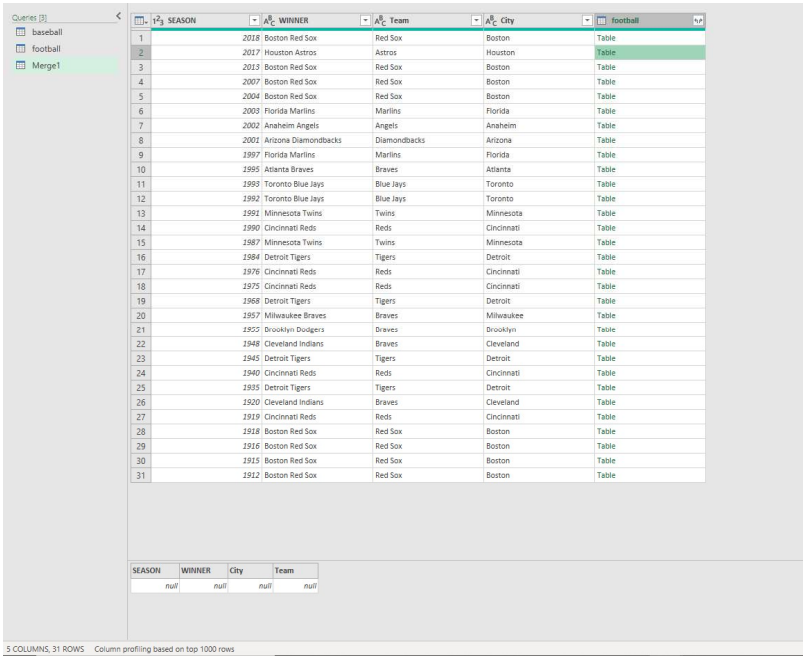
c. This is an iterative process. Power Query might get things right at first and then not later. Eventually it should get to “the truth” as determined by you. You can then click OK and use the column in your query.



0. Back to the task at hand: We want to find what teams have a baseball championship and not a football championship.
1. Open up the baseball query in the editor and go to Home > Merge Queries > Merge Queries as New.
    a. This way we don’t write over this current query, we make a new query.
    b. This will be a left anti join, to get the cities that have a baseball and not a football win.



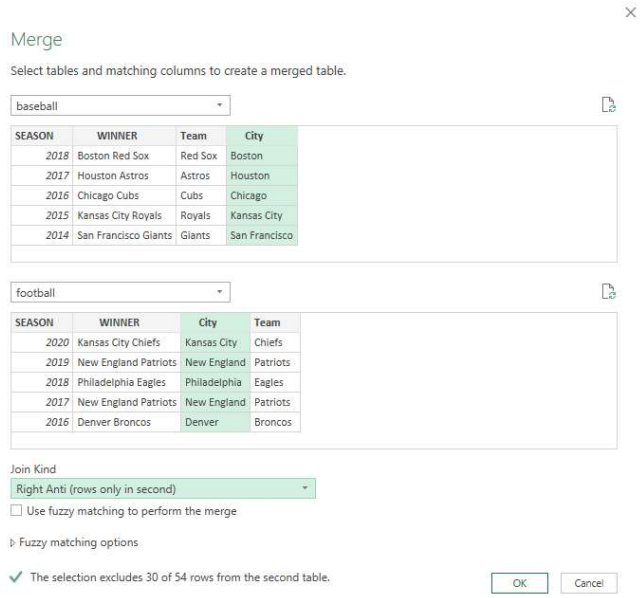
2. Click OK. You are going to see a new column “football” in our query which we can expand, however since we are only keeping the baseball records, this is going to be all blank.
    a. Since it’s a blank field, let’s delete it.



3. Here we can see all the cities that have a baseball win but not a football win. We could clean this up further if we wanted by removing the other fields and then going to Home > Remove Rows > Remove Duplicates.
4. Let’s rename this query as baseball_only.
    a. One quick thing to notice about our data – we see for example that “Florida” is listed as a city because that is the name of the team. Currently, the Florida Marlins are the Miami Marlins – and the Miami *Dolphins* have won a Super Bowl, so we could dispute whether this one should be on the list.
        i. There are lots of other ways to nitpick our results, what else can you think of?

Let’s now find cities that have a football but not a baseball win.

1. Go back to the baseball query and select Home > Merge Queries > Merge Queries as New.
2. This time we will want a right anti-join, to get only the cities with just a football championship.



3. This time it looks like we didn't get any data, however that's because all of it is "hidden" in that "football" field. Go ahead and click on it to expand. We can then get rid of the null baseball records.
    a. We now have a list of cities who have a football but no baseball championship.
    b. Let's name this query football_only.

Drill: `championships-2.xlsx`

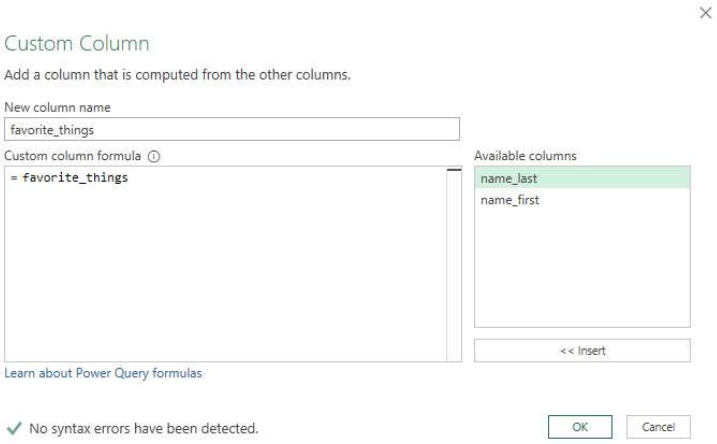
Which cities can claim *only* a hockey or basketball championship?
(Just fill out the city name, you don't need to create a team name column.)

Demo: `office-employees.xlsx`
Worksheet: `get-to-know-you`

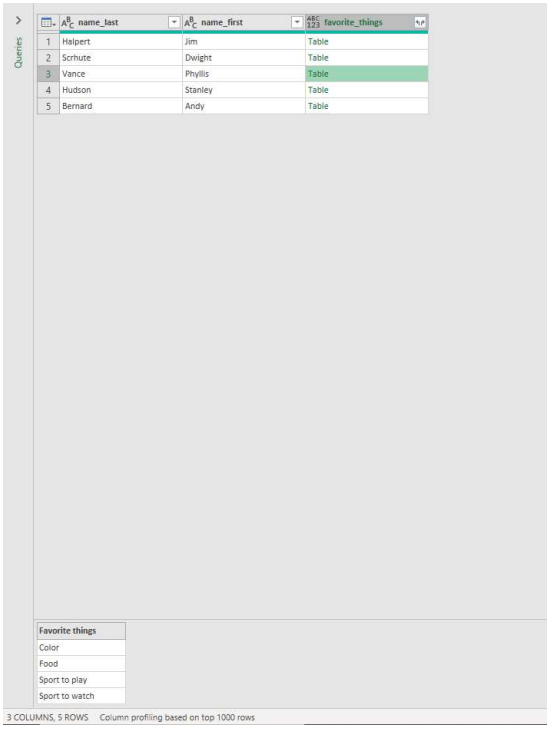
HR wants to set up a get-to-know-you activity for the sales team. You need to set up a table so that each salesperson can fill out their favorite color, food, sport to play and sport to watch.

We can do this with a cross join in Power Query:

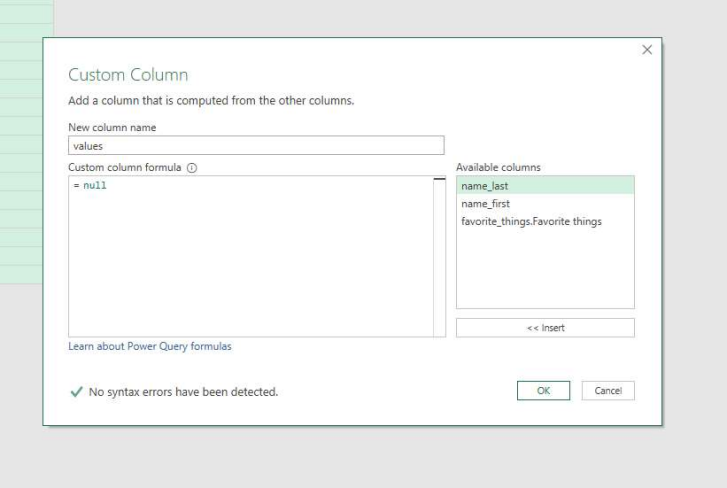
5. We've already loaded each of these tables in as queries. Click into the `names` query to edit.
6. Copy and paste the `names` query and rename it `get_to_know_you`.
7. We want to add a custom column (Add Column > Custom Column). We will name this column `favorite_things`.
    a. The formula for our column will be `favorite_things`. This is another query that shows up in the Intellisense.



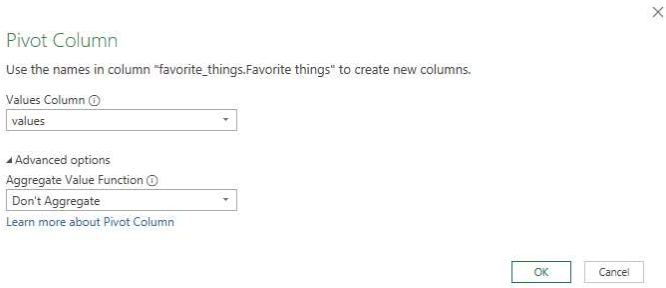
8. Click OK. Now if you click on any of the `favorite_things` cells, you can get a preview of the resulting data at the bottom of your screen:



9. Go ahead and expand the data now. We will get this in a tabular form now.
10. To pivot this table to make a checklist, we first need a “values” column to pivot on. This is blank for now so we can insert a blank or null field:



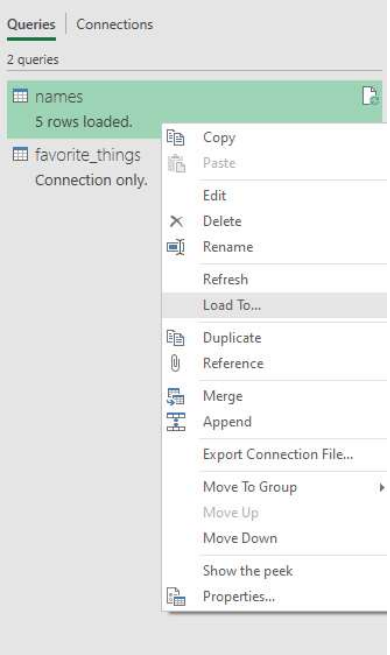
11. Now we need to "pivot" on top of `favorite_things`, based on the `values` column.
    a. Select the `favorite_things` column and go to Transform > Pivot Column.
    b. Select values as the column to pivot on, then select Advanced Options and choose "Don't Aggregate" as your aggregate value function.



12. We now have a "checklist" table that we can load into Excel.

	name_last	name_first	Color	Food	Sport to play	Sport to watch
1	Bernard	Andy	null	null	null	null
2	Halpert	Jim	null	null	null	null
3	Hudson	Stanley	null	null	null	null
4	Scrhute	Dwight	null	null	null	null
5	Vance	Phyllis	null	null	null	null

13. Currently the names query is loading to a connection only. If we want to change that we can right-click on the query and select Load To.



Drill: `states.xlsx`

Create a table to record each state's bird, flower and capital.

Demo note: Note that we can add a new property to our table, and refresh it and get that added, for example we can add the state song to the worksheet.