**APPENDING TABLES IN POWER QUERY – DEMO NOTES**

**Demo: hof\_yes.csv, hof\_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.
   1. Connect to oscars\_yes.csv
      1. Note: If we have the [direct URL to the file](https://raw.githubusercontent.com/summerofgeorge/olt-intro-to-excel-power-query/master/6-transforming-columns-ii/6-b-appending-tables/oscars_yes.csv), we can connect to the CSV file that way.
   2. 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.

Graphical user interface, table

Description automatically generated

1. Do the same thing to export oscars\_no.csv into this workbook.
2. There are now two queries in the Queries & Connections menu.
3. Right-click on the oscars\_yes query and select Append.

Graphical user interface

Description automatically generated with medium confidence

1. Now we can append oscars\_no to oscars\_yes.

Graphical user interface, application

Description automatically generated

1. This will make a *new* query, named by default Append1. Rename it to oscars\_append.
2. 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.

Graphical user interface, application

Description automatically generated

**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.)
   1. 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.

Graphical user interface, application

Description automatically generated

1. 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.

Table

Description automatically generated

1. 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.
2. 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.
3. 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.

Graphical user interface

Description automatically generated

1. 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.
   1. 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.
2. Scroll down the resulting table and we can see there are rows of null’s where there was no match for the planes data:

Table

Description automatically generated with medium confidence

1. Now we can close and load the table and I am going to name it left\_join.
2. Take the same steps except this time we will do an inner join of flights on planes.

Table

Description automatically generated

1. Another green light.
2. Same steps, expand the resulting columns and load the table.
3. 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.
4. Name the query inner\_join.

**Demo: championships.xlsx**

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

1. Preface: This data has been wrangled using Column From Examples. This is a powerful way to add a conditionally-formatted column to a table.
   1. To do this, open the football query, select WINNER field and head to the query editor and Add Column > Column From Examples > From Selection.
   2. 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.
   3. 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.

Graphical user interface, table

Description automatically generated

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

Graphical user interface, table

Description automatically generated

1. 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.
   1. Since it’s a blank field, let’s delete it.

Graphical user interface, application, table

Description automatically generated

1. 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.
2. Let’s rename this query as baseball\_only.
   1. 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.
      1. 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.

Graphical user interface, table

Description automatically generated

1. 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.
   1. We now have a list of cities who have a football but no baseball championship.
   2. Let’s name this query football\_only.