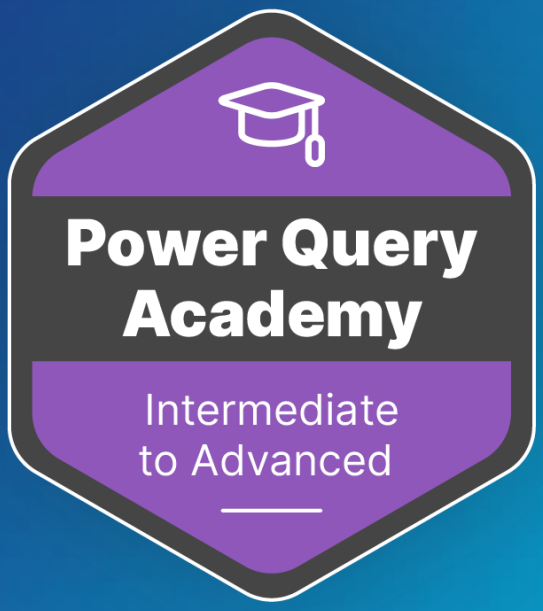


PRACTICE LAB

Grouping Data Sets

Skillwave

Table of Contents

About This Practice Lab

Copyright

How to use this Practice Lab

Our practice labs are intended to provide you the opportunity to practice pulling multiple techniques together into a near-real-world solution. They are divided into 3 sections:

1. Lab Goals

This section is a one-page flat sheet that shows you the before and after picture of the data. It is intended as an overview of your end goal for the lab. On this page, we mention some of the specific challenges that you will encounter as you work through the lab.

2. Lab Overview

The lab overview provides a quick overview of the general steps that we suggest you follow in order to complete the lab. While there are some details provided in this section, it is at a summary level only, allowing you to challenge yourself as you work through the material.

We suggest that you print this page and use it to guide yourself as you work through the data samples towards the end goal described in the Lab Goals section above.

3. Detailed Walk Through

In this section we provide the detailed and illustrated steps we took in order to create our solution. For that reason, we suggest you put it aside and ignore it until you've successfully completed the lab per the overview above. If you get stuck, or if you want to compare your solution against our approach, then this section is for you.

Keep in mind

While a completed solution is provided, this is a suggested solution only. There are doubtless many ways to solve these challenges!

Practice Lab Goals

Grouping Data Sets

We often need to manipulate data in order to report it as needed. In this case our goal is to return some summary statistics from a database that displays the percentage of gross sales by category within each major classification group. Our manager has asked to restrict the sales to the 2012 year for the "Tax Evader" location, eliminating all other years and divisions. The end product for this report should look as shown below:

Group	Category	Revenue	Group Revenue	% of Group
Food	Non Alc Beverage	1,347,619.85	1,999,700.36	67.4%
Food	Entrees	387,029.39	1,999,700.36	19.4%
Food	Appetizers	99,250.60	1,999,700.36	5.0%
Food	Sandwiches	51,649.20	1,999,700.36	2.6%
Food	Burgers	49,306.10	1,999,700.36	2.5%
Food	Breakfast	36,862.21	1,999,700.36	1.8%
Food	Soups/Salads	21,725.05	1,999,700.36	1.1%
Food	Desserts	6,257.96	1,999,700.36	0.3%
Food	Food Modifiers	-	1,999,700.36	0.0%
Alcohol	Wine	73,859.06	263,695.75	28.0%
Alcohol	Draft Beer	52,840.21	263,695.75	20.0%
Alcohol	Canned Beer	43,623.38	263,695.75	16.5%
Alcohol	Liquor	35,903.15	263,695.75	13.6%
Alcohol	Bottled Beer	32,303.55	263,695.75	12.3%
Alcohol	Coolers/Ciders	25,166.40	263,695.75	9.5%

Considerations and challenges

- Connect to a database hosted in SQL Azure
- Importing from a specific table
- Expanding records from related tables
- Reducing data
- Grouping data sets
- Performing mathematical operations

Practice Lab Overview

Suggested steps

In order to create the required output, we suggest following the route below to achieve the goal:

Importing the data set

- Open the workbook at **Grouping Data Lab - Begin.xlsx**
- Create a connection to our '*Loaded Pencil*' Microsoft Azure SQL Database
    - *HINT: Use the credentials on the Info page of the Practice Lab file*
    - *IMPORTANT: Authenticate with Database Security when prompted, NOT Windows Security*
- Select the **tlbChitDetail** table

Generating the base Revenue

- Create a column called Revenue by multiplying the Units and Amount columns

Expanding related columns

- Expand the **POSChitDate** and **tblLocations** columns from the **tblChitHeaders** column
- Expand the **Location** column from the **tblLocations** column
- Expand the **tblCategories** table from the **tblItems** column
- Expand the **POSCategoryDescription** and **POSGroup** columns from the **tblCategories** column

Reducing the data to only relevant columns and rows

- Remove the **POSChitNumber**, **POSItemCode**, **Units** and **Amount** columns
- Rename all columns to remove the "*POS*" and "*POSChit*" prefaces
- Convert the **Date** column to display only the Year
- Filter the **Date** column to **2012**
- Filter the **Location** column to "**Tax Evader**"

Generating total Group Revenue amount

- Group by **Category** and **Group** creating the following new column
    - **Revenue** which generates a Sum of the **Revenue** column

Adding the % of Total Sales share

- Group (again) by **Group**, creating the following new columns
    - **Group Revenue** which generates a Sum of the **Revenue** column
    - **Data** which uses the All Rows aggregation
- Expand the **Category** and **Revenue** columns from the **Data** column
- Create a **% of Group** column which divides the **Revenue** by the **Group Revenue**

Practice Lab Overview - Continued

Finalize the query

- Sort the **Group Revenue** and **Division** columns in Descending order
- Re-order the columns as follows: **Group**, **Category**, **Revenue**, **Group Revenue**, **% of Group**
- Set the data types for all columns as follows
    - Text: **Group, Category**
    - Currency: **Revenue, Group Revenue**
    - Percentage: **% of Group**
- Rename the query to **Sales2012**
- Close and Load the data to a table

Formatting the Excel table

- Format the columns as follows:
    - Revenue, Group Revenue: Comma Style
    - % of Group: Percentage style, 1 decimal

Detailed Walk Through

Steps to solve the challenge

Importing the data set

In order to build our solution, the first thing we need to do is connect to a Microsoft Azure hosted SQL database. To do that, you'll obviously need some credentials, which are stored in the Practice Lab file.

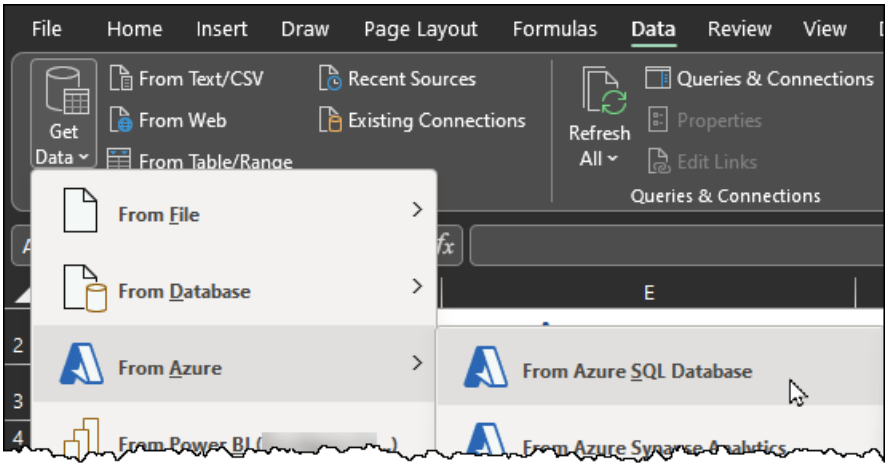
- Open the workbook at **Source File – Grouping Data Sets.xlsx**

Notice on the info page that you have the following credentials supplied:

Database Connection Info	
Server:	xlgdemos.database.windows.net
Database:	LoadedPencil

Security Credentials (Database)	
Username:	Skillwave
Password:	Sk!77w@veDemo5

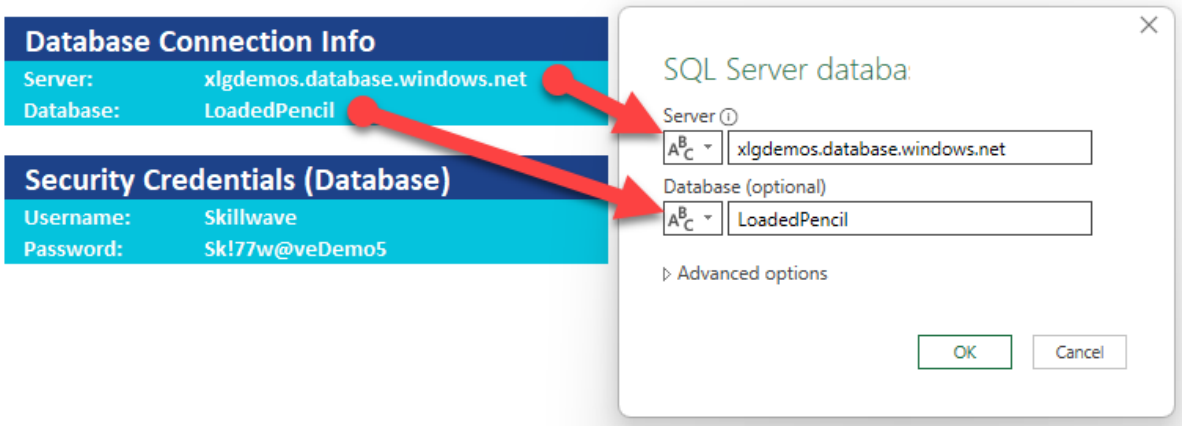
- Copy the server address in cell E6
- Go to Data → Get Data → From Azure → From Microsoft Azure SQL Database



You should now be prompted to supply the location of the server, as well as which database you'd like to connect to. Following this initial prompt you'll also get prompted for authentication if you've never connected to this database before.

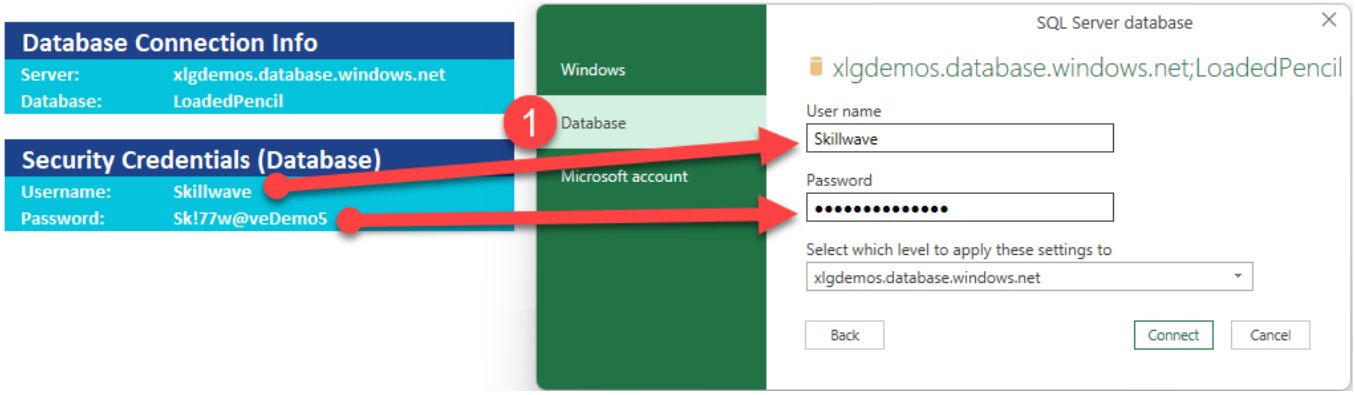
> ***HINT***
>
> *Once you've provided credentials, Power Query will store those so that you don't have to enter them every time. You can manage those under Data → Get Data → Data Source Settings*

- Complete the initial database location prompts as follows:
    - Server: **xlgdemos.database.windows.net**
    - Database: **LoadedPencil**



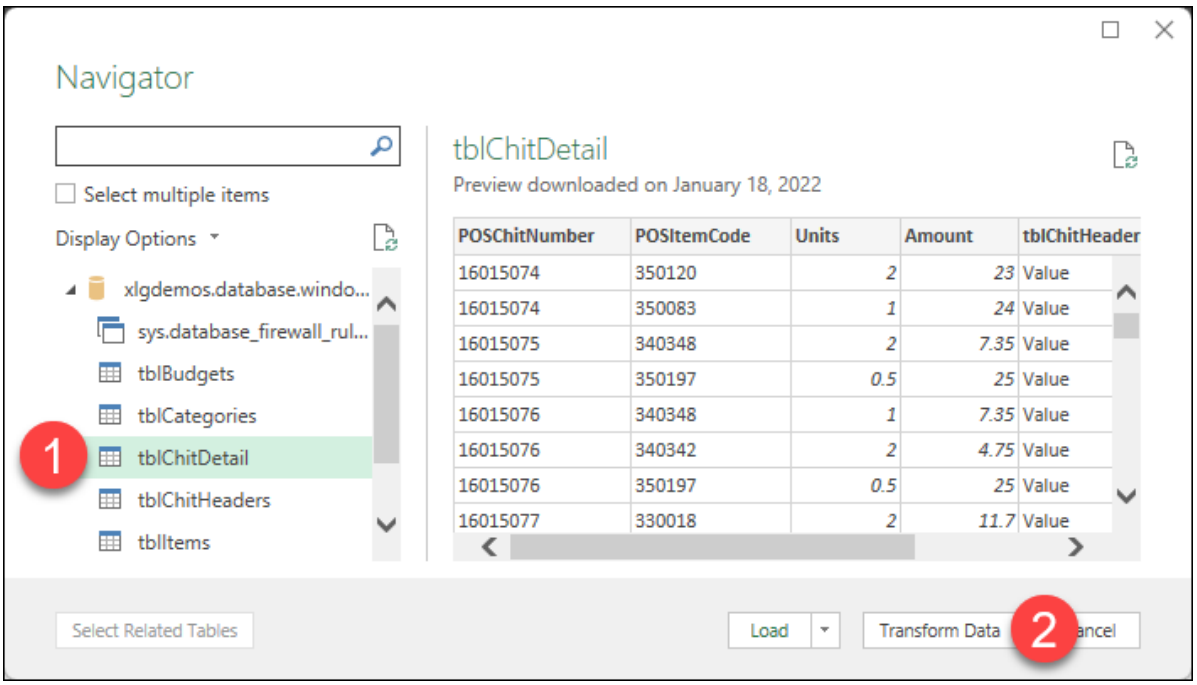
Once you click OK, you'll get prompted to enter your security information.

- Select the **Database** tab
    - Username: **Skillwave**
    - Password: **Sk!77w@veDemo5**

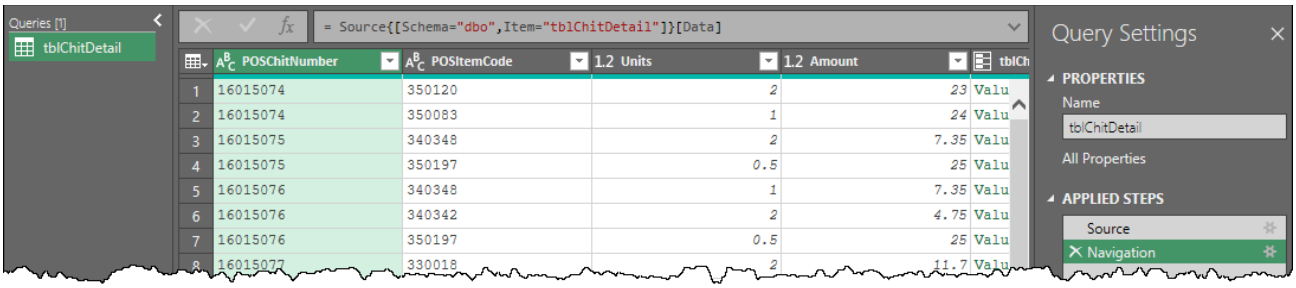


You'll now be prompted to select your table from the database.

- Select the **tblChitDetail** table from the left side
- Click the Transform Data button



You'll now be taken to the Power Query editor and, after a bit of a wait while Power Query loads a preview of the data from the web, you'll see the following:



NOTE

Depending where in the world you are, it could take a few seconds or several minutes to pull in the preview data. Patience is required when retrieving data from web hosted data sources.

Generating the base Revenue column

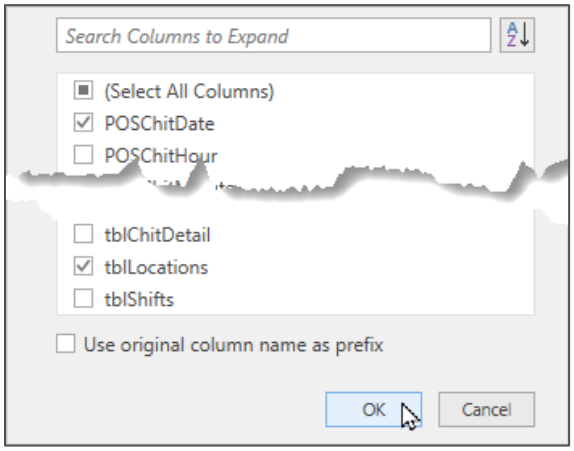
Our analysis relies heavily on computing the base Revenue, which is the product of Units x Amount. Unfortunately that column is not already present in the data set, so we will need to create it.

- Select the **Units** column
- Hold down the CTRL key and select the **Amount** column
- Go to Add Column → Standard → Multiply
- Rename the new **Multiplication** column to **Revenue**

Expanding related columns

Now that we have our base revenue, we want to extract the columns we will need to filter and group by. We'll do that via the following steps:

- At the top of the **tblChitHeaders** column click the Expand icon
    - Uncheck: **(Select All Columns)**
    - Re-check: **POSChitDate** and **tblLocations**
    - Uncheck: Use original name as column prefix



Similar to what we just did with the **tblChitHeaders** column, we want to extract some key columns from the remaining columns showing 'Value' values:

- Expand the **tblLocations** column, selecting only the **Location** column
- Expand the **tblItems** column, selecting only the **tblCategories** column
- Expand the **tblCategories** column, selecting the **POSCategoryDescription** and **POSGroup** columns

At this point, it makes sense to rename our important columns to drop the business specific terms, putting them into a more understandable English naming style.

Rename the following columns:

- **POSChitDate** column as **Date**
- **POSCategoryDescription** as **Category**
- **POSGroup** as **Group**

Now, just to make things easier, let's get rid of the columns that aren't going to be necessary for our analysis:

- Select the **Date** column → hold down the SHIFT key → select the **Revenue** column
- Right click any of the selected columns → Remove Other Columns

The output should now look similar to what is shown below:

	Date	Location	Category	Group	1.2 Revenue
1	2009-04-24 12:00:00 …	Ethical Development	Wine	Alcohol	46
2	2009-04-24 12:00:00 …	Ethical Development	Wine	Alcohol	24
3	2009-04-24 12:00:00 …	Ethical Development	Liquor	Alcohol	14.7
4	2009-04-24 12:00:00 …	Ethical Development	Wine	Alcohol	12.5
5	2009-04-24 12:00:00 …	Ethical Development	Liquor	Alcohol	7.35
6	2009-04-24 12:00:00 …	Ethical Development	Liquor	Alcohol	9.5
7	2009-04-24 12:00:00 …	Ethical Development	Wine	Alcohol	12.5
8	2009-04-24 12:00:00 …	Tax Evader	Draft Beer	Alcohol	23.4
9	2009-04-24 12:00:00 …	Tax Evader	Liquor	Alcohol	5.65
10	2009-04-24 12:00:00 …	Tax Evader	Liquor	Alcohol	169.5
11	2009-04-24 12:00:00 …	Tax Evader	Non Alc Beer	Alcohol	6

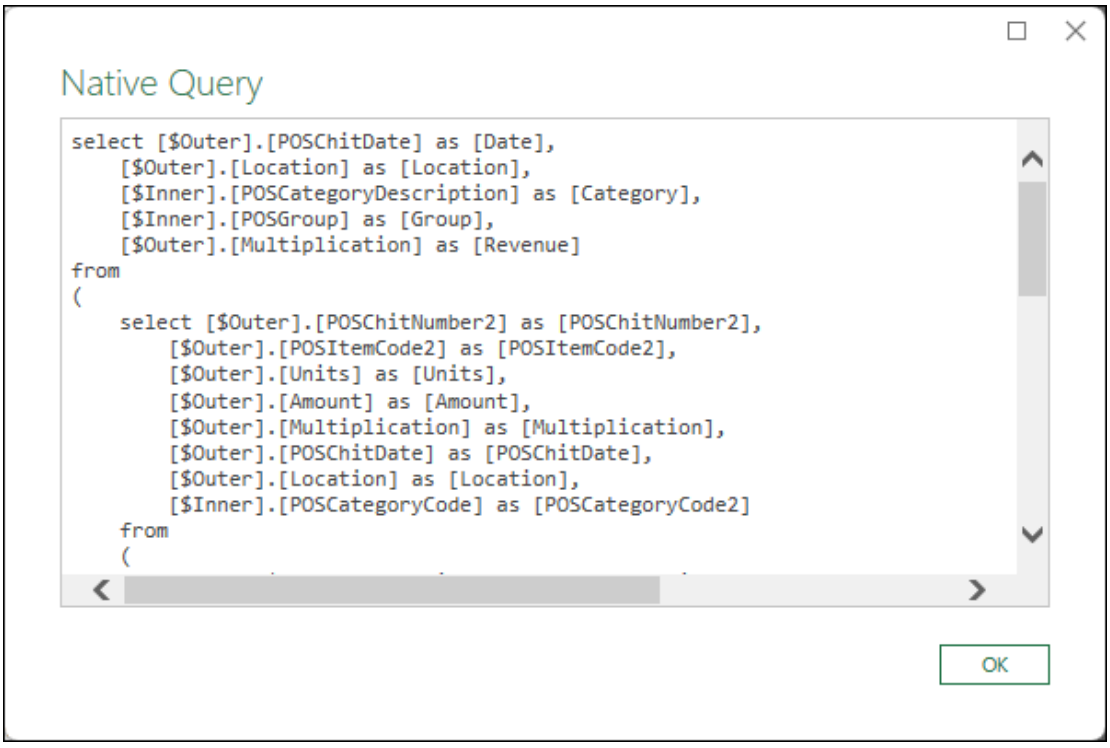
Reducing the data set to only relevant columns and rows

One of the big issues with pulling data over the web is speed. To that end we want to try and strip our data down to the smallest footprint we can, as quickly as we can. In addition, we want to be careful to not inject any custom M code (covered later in the course) which will break the Query Folding capability. The reason for this is that it sends a more efficient query to the server, letting the server process and reduce the data, sending us back a smaller data set rather than sending us all the records over the web, then letting us process them.

Before we continue it is worth checking that Query Folding is still working on our database. To do this:

- Go to the Applied Steps window (on the right side of the interface) and:
- Right click the "Renamed Columns1" step → View Native Query

You should now be presented with a big block of SQL code:



> **NOTE**
>
> *What the SQL code does isn't really important. The important part is that the code is being generated at all. If this command is greyed out, Query Folding is broken and the data will all be retrieved and processed locally. We want to keep Query Folding alive as long as possible.*

We've already removed extra columns from our data set, but let's add some filtering and reduction operations to make sure the server passes us the smallest data set possible. We will start by filtering to only records from the 'Tax Evader' division:

- Filter the **Location** column
  - Uncheck: **(Select All)**
  - Re-check: **Tax Evader**

It's now time to reduce the records to just the 2012 year, as per the manager's requirements:

- Click the calendar icon at the top left of the **Date** column → Date
- Select the **Date** column → Transform → Date → Year → Year
- Filter the **Date** column
  - Select: **Load More**
  - Type **2012** in the Search box
  - Click **OK**

> **NOTE**
>
> *After changing data types and applying filters, it is a good idea to check that query folding is still working. In this case it should be!*

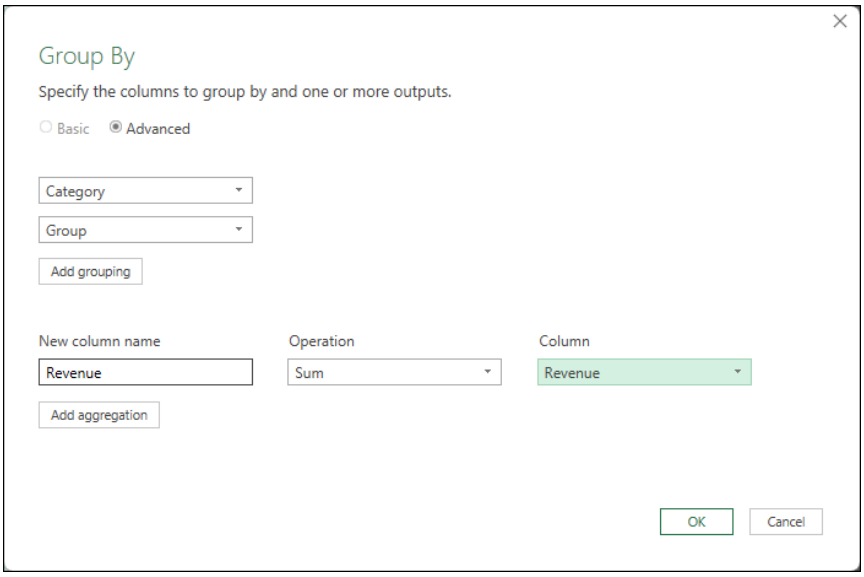
The data, at this point, should look as shown below:

	Date	Location	Category	Group	Revenue
1	2012	Tax Evader	Breakfast	Food	1.25
2	2012	Tax Evader	Appetizers	Food	0
3	2012	Tax Evader	Non Alc Beverage	Food	2
4	2012	Tax Evader	Non Alc Beverage	Food	2
5	2012	Tax Evader	Sandwiches	Food	19
6	2012	Tax Evader	Sandwiches	Food	5.95
7	2012	Tax Evader	Soups/Salads	Food	8
8	2012	Tax Evader	Sandwiches	Food	9.5
9	2012	Tax Evader	Soups/Salads	Food	9
10	2012	Tax Evader	Non Alc Beverage	Food	2

Generating the total Group Revenue amount

We are now at the point where we want to apply a grouping level to our data. The reason for this is that our records are still showing at a transactional level of detail. Since we need the total sales by category, it makes sense to reduce it.

- Select the **Group** column → hold down the CTRL key → select the **Category** column
- Go to the Transform tab → Group By → modify the "New Column Name" as follows:
  - New column name: Revenue
  - Operation: Sum
  - Column: Revenue



The results are drastically different than our original data set. Instead of retrieving over 300,000 records, we are down to only the 15 rows we need:

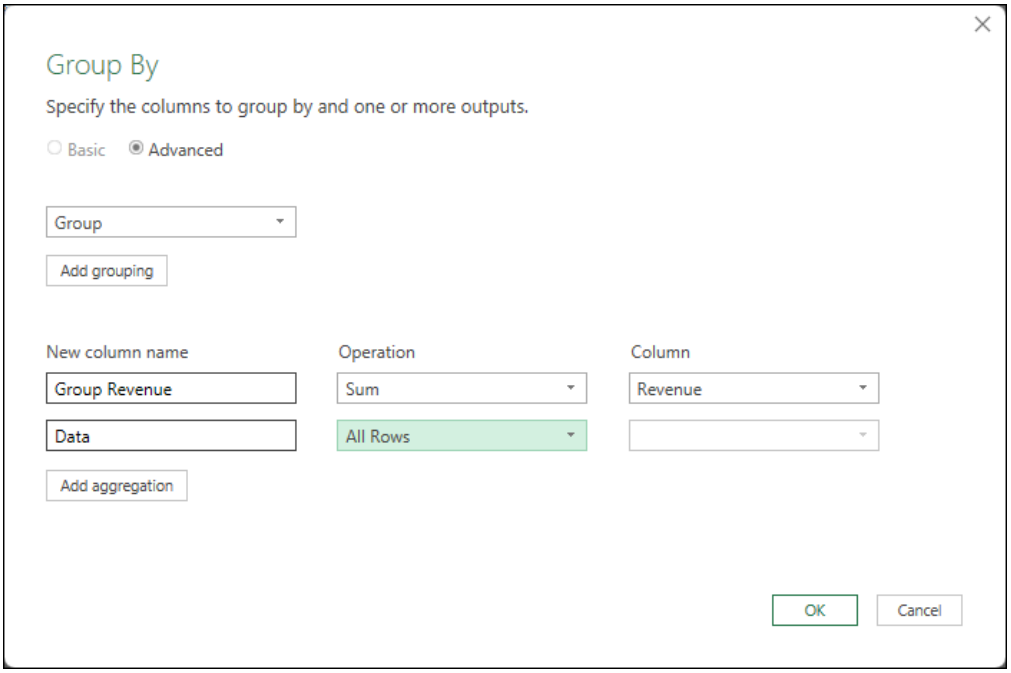
	Category	Group	Revenue
1	Bottled Beer	Alcohol	32303.55
2	Canned Beer	Alcohol	43623.38145
3	Coolers/Ciders	Alcohol	25166.4
4	Draft Beer	Alcohol	52840.20785
5	Liquor	Alcohol	35903.15
6	Wine	Alcohol	73859.0649
7	Appetizers	Food	99250.6022
8	Breakfast	Food	36862.2125
9	Burgers	Food	49306.1
10	Desserts	Food	6257.9609
11	Entrees	Food	387029.3875
12	Food Modifiers	Food	0
13	Non Alc Beverage	Food	1347619.85
14	Sandwiches	Food	51649.2008
15	Soups/Salads	Food	21725.05

In addition, you'll notice that:

- The columns we did not select in the grouping are no longer present, and
- Query folding is *still* working!

Now we want to try and generate the total revenue by Group. To do this, we will need to leverage another grouping:

- Select the Group column → go to the Transform tab → Group By → Advanced
- Create the following new columns
    - **Group Revenue** which generates a Sum of the **Revenue** column
    - **Data** which uses the All Rows operation (click *Add aggregation* to add a row)



Our table is much shorter, but we have managed to generate the total amount of revenue by Group, as shown below.

	Group	Group Revenue	Data
1	Alcohol	263695.7542	Table
2	Food	1999700.364	Table

Let's extract the original data that was used for the grouping, but before we do, we are going to make one quick change, and re-order the columns:

- Move the Data column between the Group and Group Revenue columns

	Group	Data	Group Revenue
1	Alcohol	Table	263695.7542
2	Food	Table	1999700.364

NOTE

The reason we took this step is that we want to push as many commands as possible into the query folding capabilities. And if you check, you'll see that the query folding is STILL active at this point!

Adding the % of Total Sales share

Now it's time to restore our original data…

- Click the Expand icon at the top of the Data column
- Uncheck Group (as we already have that data)

The result is that we have now added a Group Revenue column to our original data, which returns the total revenue by Group:

	Group	Category	Revenue	Group Revenue
1	Alcohol	Bottled Beer	32303.55	263695.7542
2	Alcohol	Canned Beer	43623.38145	263695.7542
3	Alcohol	Coolers/Ciders	25166.4	263695.7542
4	Alcohol	Draft Beer	52840.20785	263695.7542
5	Alcohol	Liquor	35903.15	263695.7542
6	Alcohol	Wine	73859.0649	263695.7542
7	Food	Appetizers	99250.6022	1999700.364
8	Food	Breakfast	36862.2125	1999700.364
9	Food	Burgers	49306.1	1999700.364
10	Food	Desserts	6257.9609	1999700.364
11	Food	Entrees	387029.3875	1999700.364
12	Food	Food Modifiers	0	1999700.364
13	Food	Non Alc Beverage	1347619.85	1999700.364
14	Food	Sandwiches	51649.2008	1999700.364
15	Food	Soups/Salads	21725.05	1999700.364

With this column in place, we can now calculate the percentage of revenue by group:

- Select the **Revenue** column → hold down the CTRL key → select the **Group Revenue** column
- Go to Add Column → Standard → Divide
- Rename the Division column as **% of Group**

Finalizing the Query

The only transformation we have left to do is sort the data correctly before loading it to Excel. To be fair, you could do this in Excel after loading the data, but by doing it in Power Query, we can make sure that the data shows up in the right order without needing any manual intervention to correct it upon a refresh. Let's set up a sorting order as follows:

- Click the filter arrow at the top right of the **Group Revenue** column → Sort Descending
- Click the filter arrow at the top right of the **% of Group** column → Sort Descending

> ***NOTE***
>
> *If you check the Applied Steps window, you'll notice query folding is no longer working. In fact, it was actually the expansion of the Data column which finally broke it. Everything from that point forward will therefore be processed with local CPU and RAM resources. The good news is that we don't have much left to do!*

The final thing we should do before we load our query is lock-in our data types and name our query.

- Select the **Group** and **Category** columns → Transform → Data type → Text
- Select the **Revenue** and **Group Revenue** columns → Transform → Data type → Currency
- Select the **% of Group** column → Transform → Data type → Percentage
- Rename the query to **Sales2012**

Our query is now complete:

	Group	Category	Revenue	Group Revenue	% of Group
1	Food	Non Alc Beverage	1,347,619.85	1,999,700.36	67.39%
2	Food	Entrees	387,029.39	1,999,700.36	19.35%
3	Food	Appetizers	99,250.60	1,999,700.36	4.96%
4	Food	Sandwiches	51,649.20	1,999,700.36	2.58%
5	Food	Burgers	49,306.10	1,999,700.36	2.47%
6	Food	Breakfast	36,862.21	1,999,700.36	1.84%
7	Food	Soups/Salads	21,725.05	1,999,700.36	1.09%
8	Food	Desserts	6,257.96	1,999,700.36	0.31%
9	Food	Food Modifiers	0.00	1,999,700.36	0.00%
10	Alcohol	Wine	73,859.06	263,695.75	28.01%
11	Alcohol	Draft Beer	52,840.21	263,695.75	20.04%
12	Alcohol	Canned Beer	43,623.38	263,695.75	16.54%
13	Alcohol	Liquor	35,903.15	263,695.75	13.62%
14	Alcohol	Bottled Beer	32,303.55	263,695.75	12.25%
15	Alcohol	Coolers/Ciders	25,166.40	263,695.75	9.54%

NOTE

Before we take the final step of loading our data, it is worth noting that query folding is broken, as evidenced by the fact that the View Native Query command is greyed out. In fact, if you go back step by step, you'll find that the "Expanded Data" step, immediately after we re-ordered the grouped table, broke query folding. From that point forward, everything must be processed with local CPU and RAM, instead of asking SQL Server to do the heavy lifting. This is the reason why the re-ordering of the columns was done prior to expansion. Had we waited, that operation would have needed to be processed locally as well.

It's finally time to load this to a worksheet. Go to File → Close & Load

Formatting the Excel table

The last thing we need to do for our solution is fix the number formatting since Power Query doesn't push out percentage or comma style formats:

- Format the columns as follows:
  - Columns C, D: Comma Style
  - Column E: Percentage style, 1 decimal

And the end result will appear as follows:

	A	B	C	D	E
1	Group	Category	Revenue	Group Revenue	% of Group
2	Food	Non Alc Beverage	1,347,619.85	1,999,700.36	67.4%
3	Food	Entrees	387,029.39	1,999,700.36	19.4%
4	Food	Appetizers	99,250.60	1,999,700.36	5.0%
5	Food	Sandwiches	51,649.20	1,999,700.36	2.6%
6	Food	Burgers	49,306.10	1,999,700.36	2.5%
7	Food	Breakfast	36,862.21	1,999,700.36	1.8%
8	Food	Soups/Salads	21,725.05	1,999,700.36	1.1%
9	Food	Desserts	6,257.96	1,999,700.36	0.3%
10	Food	Food Modifiers	-	1,999,700.36	0.0%
11	Alcohol	Wine	73,859.06	263,695.75	28.0%
12	Alcohol	Draft Beer	52,840.21	263,695.75	20.0%
13	Alcohol	Canned Beer	43,623.38	263,695.75	16.5%
14	Alcohol	Liquor	35,903.15	263,695.75	13.6%
15	Alcohol	Bottled Beer	32,303.55	263,695.75	12.3%
16	Alcohol	Coolers/Ciders	25,166.40	263,695.75	9.5%