



# Introduction to Power Query

For Data Essentials L3

[qa.com](http://qa.com)



# Contents

---

<b>1</b>	<b>Introduction .....</b>	<b>3</b>
1.1	Learning outcomes.....	3
1.2	Prerequisites .....	3
1.3	Completing the activities .....	4
1.4	Power Query in different versions of Excel .....	5
<b>2</b>	<b>Queries .....</b>	<b>6</b>
2.1	What is a Query? .....	6
2.2	Creating a Query.....	7
2.3	Example: import a CSV file.....	8
2.4	Summary.....	12
<b>3</b>	<b>Getting into and out of Power Query .....</b>	<b>12</b>
3.1	How to open Power Query from Excel.....	12
3.2	How to close Power Query and get back to Excel.....	13
3.3	How to toggle between Table and Connection .....	14
3.4	Another way to switch to Connection mode .....	17
3.5	Summary.....	18
<b>4</b>	<b>Navigating Power Query .....</b>	<b>19</b>
4.1	The Power Query Editor window .....	19
4.2	Queries panel .....	19
4.3	Data preview area .....	21
4.4	Applied Steps .....	22
	Creating a Step .....	24
	Navigating Steps.....	24
	Deleting a Step .....	25
	Editing a Step.....	26
4.5	Summary.....	26
<b>5</b>	<b>Basic transformations .....</b>	<b>27</b>
5.1	Use first row as headers.....	27
5.2	Remove the secondary header row.....	29
5.3	Add an index column .....	30
5.4	Change column types.....	31



5.5	Remove empty rows .....	35
5.6	Arithmetic.....	39
5.7	Rounding.....	41
5.8	Split a column.....	42
5.9	Find and replace .....	44
5.10	Summary.....	45
<b>6</b>	<b>Merging Queries .....</b>	<b>45</b>
6.1	Create lookup tables.....	47
6.2	Import lookup tables into Power Query .....	50
6.3	Merge lookup tables .....	54
6.4	Summary.....	60
<b>7</b>	<b>Column From Examples.....</b>	<b>60</b>
7.1	Extract weekday name.....	61
7.2	Custom date format.....	64
7.3	Surnames.....	65
7.4	Multiple source columns.....	66
7.5	Apply to student_grades.....	68
7.6	Summary.....	69
<b>8</b>	<b>Group and aggregate .....</b>	<b>69</b>
8.1	Is study time predictive of grade?.....	69
8.2	Is having a romantic partner predictive of grade?.....	71
<b>9</b>	<b>Help and support.....</b>	<b>72</b>
<b>10</b>	<b>Appendices.....</b>	<b>73</b>
	Appendix 1: Notes on the Data Model .....	73



# 1 Introduction

**Power Query** is a data transformation engine built into Microsoft Excel and Microsoft Power BI. With Power Query, we can create pipelines for retrieving, cleaning, and transforming data. This type of processing is often referred to as ETL (Extract, Transform, Load).

This guide introduces the main features of the Power Query Editor in Excel.



## Useful Tool

Power Query is a fast, versatile, non-destructive data transformation tool. The combined capabilities of Excel and Power Query are sufficient to complete most technical activities in this apprenticeship.

Excel and Power Query are also suitable tools to use for the Scenario Demonstration exam at EPA (End-Point Assessment).

## 1.1 Learning outcomes

In this guide, we will cover:

- How to open and close the Power Query Editor.
- The layout and core features of Power Query.
- How to apply some of the most common data transformations.

## 1.2 Prerequisites

- One of the following **desktop app** versions of Excel:
  - Excel 365 for Windows.
  - Excel 2016 or later for Windows.
  - Excel 365 for Mac v16.69 or later.



- Basic familiarity with Excel is beneficial.



## Warning

At the time of this writing, Power Query is available in the Excel **desktop app** only. Power Query is not available in Excel for the Web or the Excel mobile app.

## 1.3 Completing the activities

This guide includes **Activity** steps. If you have not used Power Query before, it is recommended that you complete these brief activities to gain hands-on experience with Power Query



## Activity

Activities look like this!

The activities align with the screenshots.

Additionally, Sections 5–7.6 contain a series of worked examples of various features of Power Query. If you would like to follow along with these examples, make sure you have a copy of the data files below.

- “student\_grades.csv”
- “column\_from\_examples.csv”

If you downloaded this guide from Cloud Academy in a zip file, these files were contained within the same zip file. You should **extract** the contents of the zip file before you continue. (In Windows, **Shift + right click > Extract All....**)

If a QA tutor sent you this guide as a standalone PDF, the tutor can provide you the files separately.



## 1.4 Power Query in different versions of Excel

---

Power Query (also called **Get & Transform**) is available in the versions of Excel indicated below.

### Excel for Windows

Office 2013 and earlier	X
Office 2016	✓
Office 2019	✓
Office 2021 and later	✓
Microsoft 365 (all versions)	✓

### Excel for Mac OS

Office 2016 for Mac and earlier	X
Office 2019 for Mac	X
Microsoft 365 for Mac v16.68 and earlier	X
Microsoft 365 for Mac v16.69 (released Jan 2023) and later	✓

### Excel for the Web

Excel for the Web does not include the Power Query Editor. The feature set of Excel for the Web is also extremely limited compared to the Excel desktop app. As such, Excel for the Web is **not recommended** for this apprenticeship.

### Excel mobile app

The Excel mobile app has many of the same limitations as Excel for the web. As such, the Excel mobile app is **not recommended** for this apprenticeship.



## Helpful Hint

Each version of Excel is slightly different. If a step requires a significantly different approach in different versions of Excel, look for version hints such as **Excel 365** and **Excel 2016**. Note: these versions refer to Excel for Windows.



## Warning

The screenshots in this guide are of Excel 365 for Windows. If you are using a different version of Excel, please note that your UI may look slightly different.

## 2 Queries

### 2.1 What is a Query?

In Power Query, a **Query** can be thought of as a pointer to a data source. Additionally, a Query may apply a “recipe” that transforms the data. The output of a Query is a **Table**.

A Query describes:

- The input data source. Many different types of data source are supported, including files, databases, and web pages.
- What transformations to apply to the input data.

For example, a simple Query might perform the following steps:

- 1) Load source file: “C:\mydata.csv”.
- 2) Delete column “Column1”.



3) Convert column “Column2” to type “Whole number”.

The output of a Query is a **Table**. Typically, this output Table is loaded to an Excel spreadsheet.



### Key Point

Queries can be edited at any time. Both the data source and the transformation “recipe” can be edited.



### Key Point

A Query **points** to source data. If the source data is later modified, the Query will pick up the changes the next time the Query is refreshed.

## 2.2 Creating a Query

Creating a Query in Excel is simple and intuitive. To begin, we specify the type of data source for the Query.

1. In Excel, navigate to:

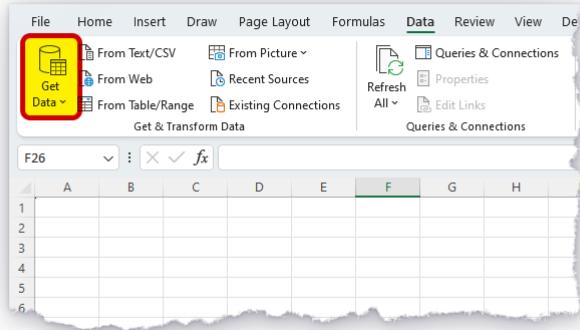
Excel 2019+ | Excel 365

Ribbon > Data tab > Get & Transform Data group > Get Data.

Excel 2016

Ribbon > Data tab > Get & Transform group > New Query.

2. A dropdown menu appears. Now, we simply click the type of source that we want. For example, CSV, Excel workbook, or SQL Server.



**Figure 1:** The **Get Data** menu allows us to choose from many types of data source.

## 2.3 Example: import a CSV file

As an example, we will import the provided CSV file “student\_grades.csv”.



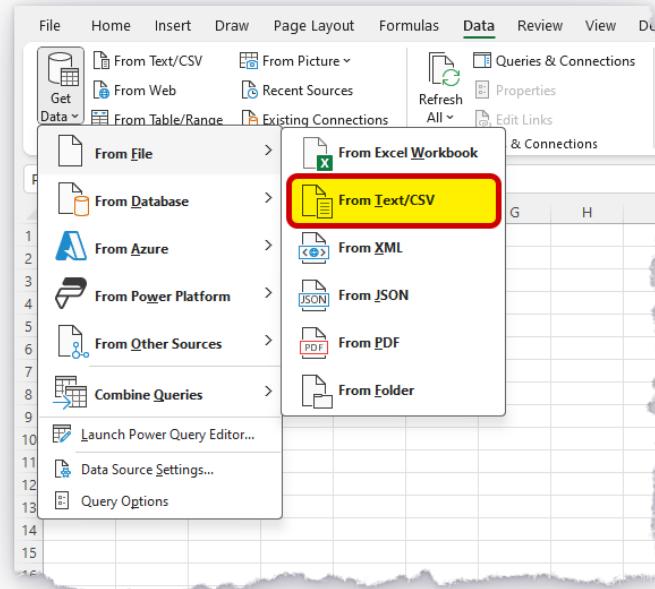
### Important Idea or Concept

We will use “student\_grades.csv” throughout this guide. If you wish to follow along with the activities and compare your solutions with the screenshots, you should import “student\_grades.csv” now.

1. Navigate to:

Excel 2019+ | Excel 365

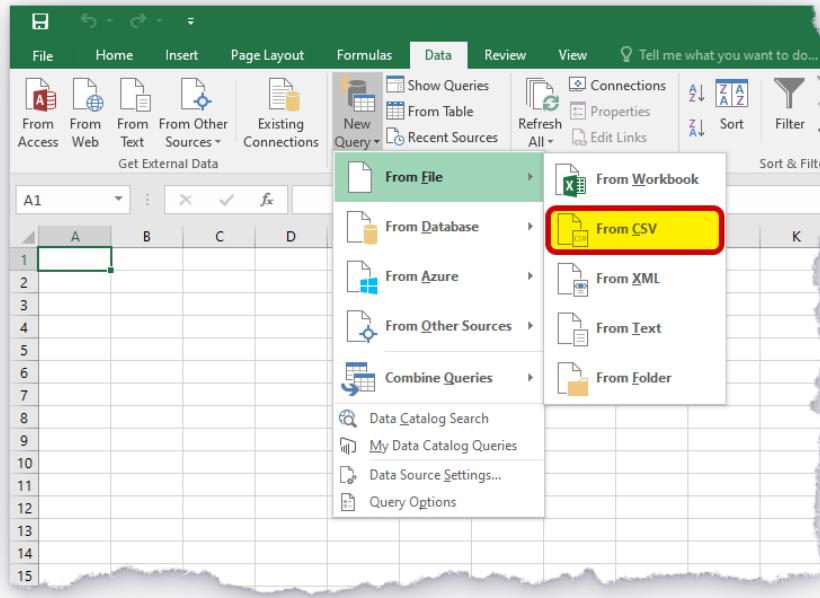
Ribbon > Data tab > Get & Transform Data group > Get Data > From File > From Text/CSV.



**Figure 2:** Importing data from a text or CSV file in Excel 2019+ or Excel 365.

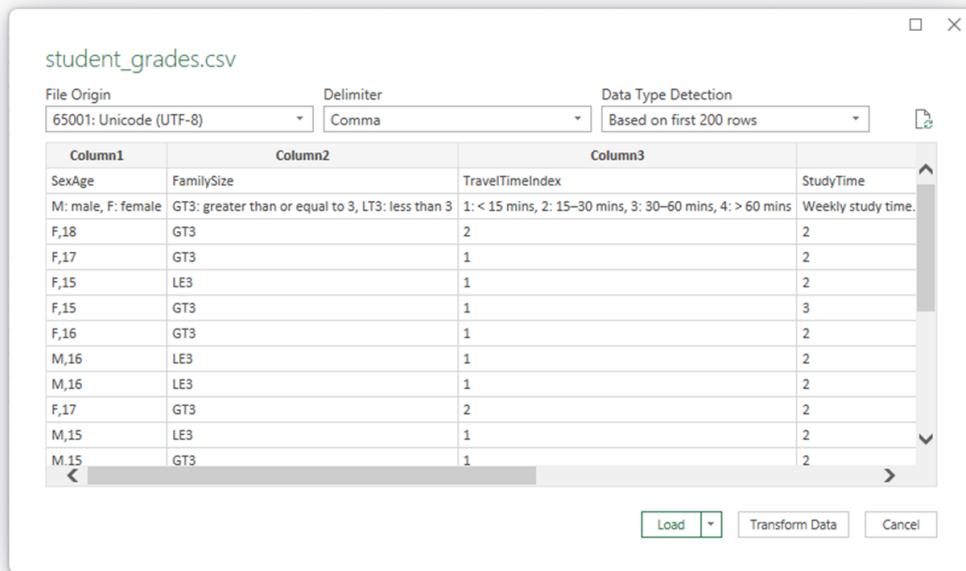
**Excel 2016**

Ribbon > **Data** tab > **Get & Transform** group > **New Query** > **From File** > **From CSV**.



**Figure 3:** Importing data from a CSV file in Excel 2016.

2. The **Import Data** dialog appears. We will import "student\_grades.csv".
3. The **data preview** appears. This window displays a faithful preview of the CSV file. The only transformation that has been applied is the parsing of commas as column breaks.



**Figure 4:** The data preview for a CSV file.



- At this point, we can click either **Load** or **Transform Data**.

- **Load**

The CSV file will be loaded directly to an Excel spreadsheet as a **Table** linked to a **Query**.

- **Transform Data**

This also loads the CSV file as a **Query**. Additionally, we are taken to the **Power Query Editor** window, where we can apply transformations to the Query prior to loading to Excel.

The only difference between the two options is that **Transform Data** will open the Power Query Editor. However, we are free to open the Power Query Editor any time we like. So, it **doesn't really matter** which button we click.

Let's click **Load**.

5. The Navigator window closes, and we are taken back to Excel. The **Queries & Connections** panel should now be visible on the right-hand side of the screen.

The screenshot shows a Microsoft Excel interface with the 'student\_grades' query listed in the 'Queries & Connections' panel on the right. The main area displays a table with three columns: Column1, Column2, and Column3. The data includes rows such as 'SexAge', 'FamilySize', and 'TravelTimeIndex'. The 'Queries & Connections' panel shows '1 query' named 'student\_grades' with '397 rows loaded.'

Column1	Column2	Column3
SexAge	FamilySize	TravelTimeIndex
M: male, F: female	GT3: greater than or equal to 3, LT3: less than 3	1: < 15 mins, 2: 15–30 mins, 3: 30–60 mins, 4: > 60 min
F,18	GT3	2
F,17	GT3	1
F,15	LE3	1
F,15	GT3	1
F,16	GT3	1
M,16	LE3	1
M,16	LE3	1
F,17	GT3	2
M,15	LE3	1
M,15	GT3	1
F,15	GT3	1
F,15	GT3	3
M,15	LE3	1
M,15	GT3	2

**Figure 5:** The Queries & Connections panel displays the Query we just created.



If the Queries & Connections panel is not visible, go to **Ribbon** > **Data** tab > **Queries & Connections** group. Toggle the **Queries & Connections** button.

6. We have now loaded “student\_grades.csv” as an Excel Table linked to a Query. Verify that your Table has 7 columns and 397 rows.

## 2.4 Summary

---

- A Query is a kind of “recipe” that points to a data source and applies transformations to it.
- The output of a Query is a Table.
- The **Data** tab of the Ribbon provides buttons for creating a Query.

We have imported “student\_grades.csv” as a Query. However, we can see that the resultant **student\_grades** Table has a few irregularities. We will fix those later. But first, we will learn how to navigate Power Query.

## 3 Getting into and out of Power Query

---



### Key Point

While the Power Query Editor window is open, editing in Excel is blocked.

### 3.1 How to open Power Query from Excel

---

There are several methods of opening Power Query from Excel.

#### Method 1

Press **Alt + F12**. On some keyboards: **Alt + Fn + F12**.



## Method 2

Navigate to **Ribbon** > **Data** tab > **Get & Transform Data** group > **Get Data** > **Launch Power Query Editor....**

## Method 3

In the **Queries & Connections** panel, **right-click** any Query and click **Edit**.

## Method 4

**Excel 365**

In the **Search Bar** (**Alt + Q**), start typing “Power Query Editor”. Click the autocomplete suggestion “Launch Power Query Editor”.

**Note:** It is possible to open the Power Query Editor even if you have not yet created any Queries.



### Helpful Hint

The Power Query Editor can also be opened by clicking **Transform** when creating a new Query. However, it is not necessary to create a new Query every time you want to open the Power Query editor!

## 3.2 How to close Power Query and get back to Excel

To close Power Query, go to **Ribbon** > **Home** tab > **Close & Load**.

Notice that we have two options: **Close & Load** and **Close & Load To...**. The second option, **Close & Load To...**, lets us choose whether to load a Query as an Excel Table or as a Connection. If we choose Connection, the Query will be loaded in memory only.



**Figure 6:** The **Close & Load** button in Power Query.

We can freely change the “Load To” location of a Query at any time. Therefore, it **doesn't matter** which option we choose.



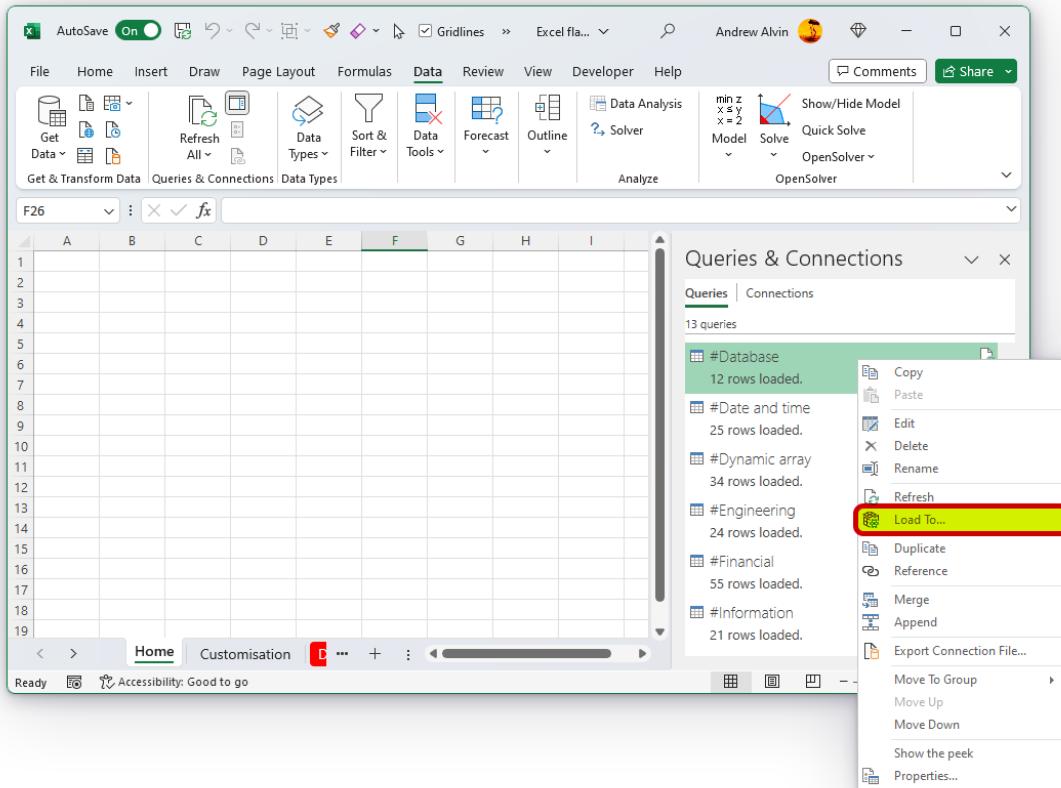
### Activity

We have seen four different ways of opening the Power Query Editor window. Try each of these methods at least once. Practise your preferred method until you can open and close Power Query fluently.

## 3.3 How to toggle between Table and Connection

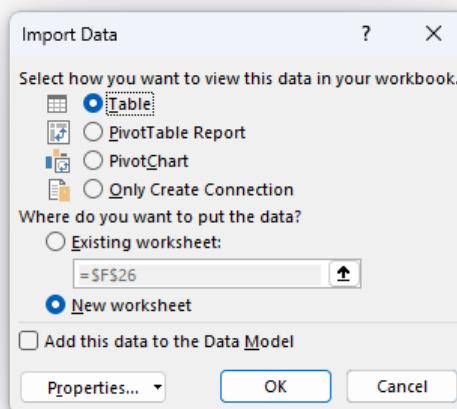
As mentioned earlier, a Query can be loaded as a Table or a Connection. To change a Query's load destination, follow these steps.

1. Open the **Queries & Connections** panel > **right-click** a Query > **Load To...**



**Figure 7:** How to change the load destination of a Query.

2. The **Import Data** window appears.



## Table

Load the output of the Query to an Excel Table.



### PivotTable Report

*Not needed in this apprenticeship.*

### PivotChart

*Not needed in this apprenticeship.*

### Only Create Connection

Load the output of the Query in memory only.

### Add this data to the Data Model

*Not needed in this apprenticeship. You do not need to check this box, but it is harmless to do so.*



### Helpful Hint

The most common settings we will need are:

- Load to: **Table**.
- Destination: **New worksheet**.



### Helpful Hint

A Query loaded as a Table can be distinguished from a regular Excel Table in a few ways:

- The default colour of a Query Table is green, while a regular Table is blue.
- When you click a cell in the Table, check the Ribbon. If the Table is a Query Table, a tab called “Query” will appear on the far right.
- In the Queries & Connections panel, if you click a Query that is loaded as a Table, the Table will become selected and focused.



## Activity

Toggle the load destination of some of your Queries and observe the effects.

### 3.4 Another way to switch to Connection mode

If you wish to change a Query's load destination from **Table** to **Connection**, one quick method is to simply delete the worksheet that the Query is loaded to, as shown below.

The screenshot shows a Microsoft Excel window with the Power Query ribbon selected. A context menu is open over a query named 'DAVERAGE' in the 'Purpose' column. The 'Delete' option in the menu is highlighted with a red box. To the right of the menu, the 'Queries & Connections' pane is visible, showing a list of 13 queries, including '#Database' which has 12 rows loaded.

Deleting the worksheet does not delete the Query, it merely converts the Query's load destination from Table to Connection.



## Warning

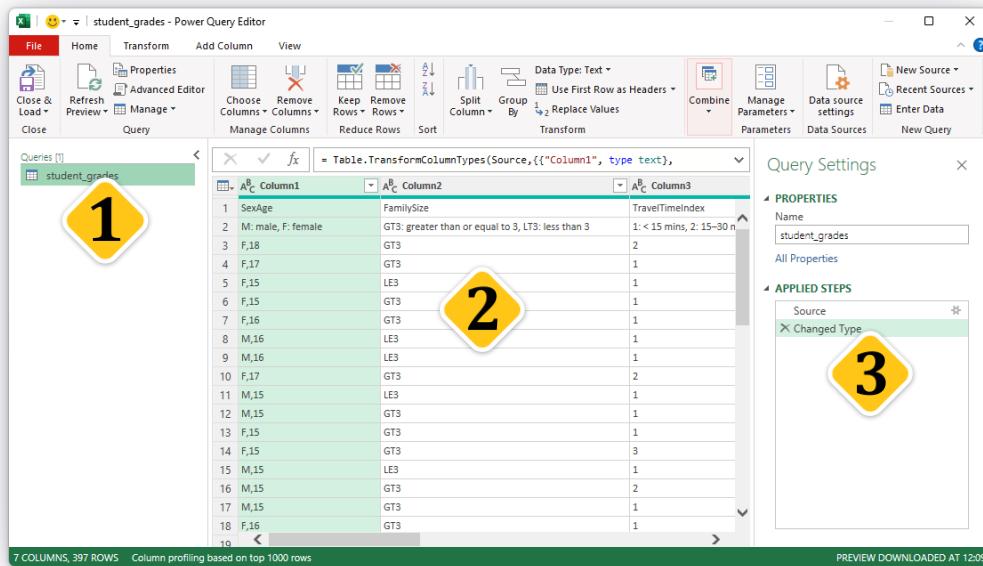
Be careful whenever you delete a worksheet, as the action is not undoable. Make sure there is no important data on the worksheet. It is good practice to **back up** your work before any irreversible operation.

### 3.5 Summary

- There are several different ways of opening Power Query from Excel.
- While the Power Query Editor window is open, editing in Excel is blocked.
- The Applied Steps list is a navigable, editable history of all transformations applied within a Query.
- Many transformations in Power Query can be achieved simply by clicking buttons on the Ribbon.
- The load destination of a Query can be toggled between Table and Connection at any time

## 4 Navigating Power Query

### 4.1 The Power Query Editor window



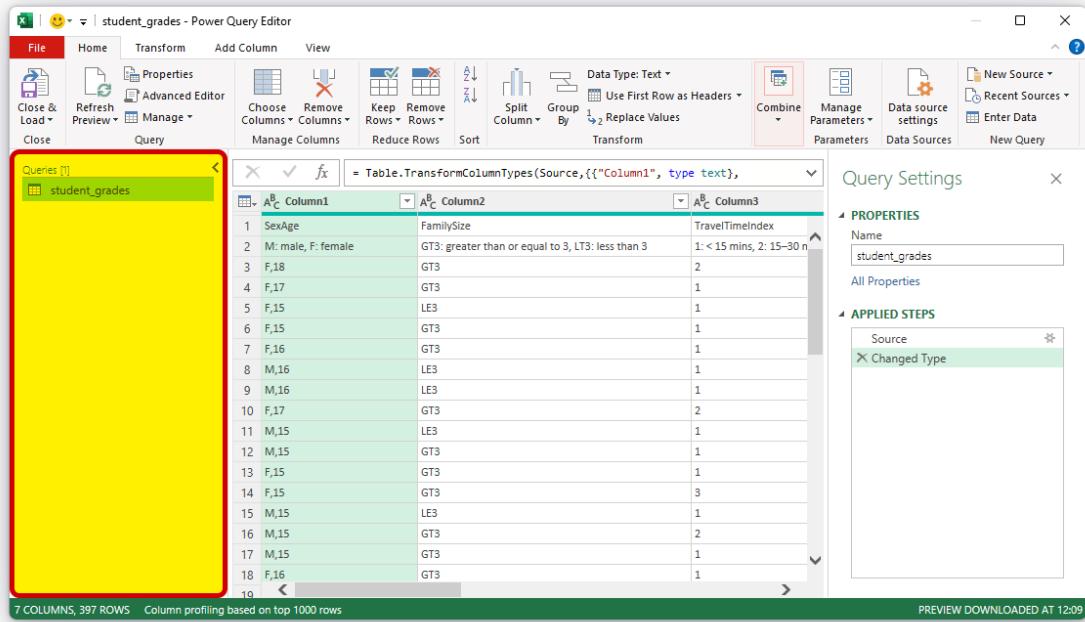
**Figure 8:** The Power Query Editor window.

- 1) The **Queries** panel: a navigable list of all Queries in this workbook.
- 2) The **data preview** area. Note: the table data cannot be modified by manually typing into the cells. In Power Query, the data set is modified by applying transformations to rows and columns.
- 3) The **Applied Steps**: an ordered list of data transformations that have been applied to the source data. This is our data transformation “recipe”.

**Important:** note the **Close & Load** button in the top left. This is how we close Power Query and get back to Excel.

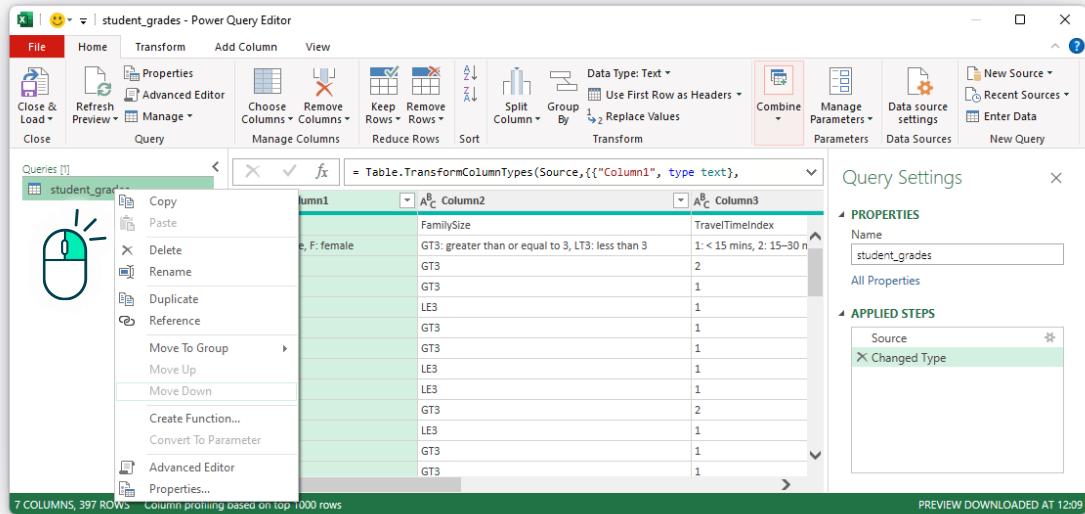
### 4.2 Queries panel

The **Queries** panel displays a list of all Queries in this workbook.



**Figure 9:** The Queries panel.

**Right-click** a Query to bring up a menu of useful commands such as **Delete**, **Rename**, **Duplicate**, and **Reference**.



**Figure 10:** In the Queries panel, right-click a Query to bring up a menu of useful commands.



## Helpful Hint

If you are about to make destructive edits to one of your Queries, it is a good idea to **duplicate** it so you have a backup.

If we **right-click** the empty space in the Queries panel and navigate to **New Query**, we can create a new Query the same way we did from Excel.

### 4.3 Data preview area

The **data preview** area displays a preview of the currently selected Query at the currently selected **Step**. (**Steps** are explained in the next section.)

The screenshot shows the Power Query Editor interface. The 'student\_grades' query is selected in the Queries list. The main area displays a preview of the data with three columns: SexAge, FamilySize, and TravelTimelIndex. The preview pane shows 18 rows of data. The 'Changed Type' step is highlighted in the 'APPLIED STEPS' pane. The status bar at the bottom indicates 7 COLUMNS, 397 ROWS and PREVIEW DOWNLOADED AT 1209.

**Figure 11:** The data preview area.

Data in the data preview cannot be edited directly. Power Query is not a tool for manual data entry: it is a tool for applying transformations and formulas to entire columns, rows, or tables.

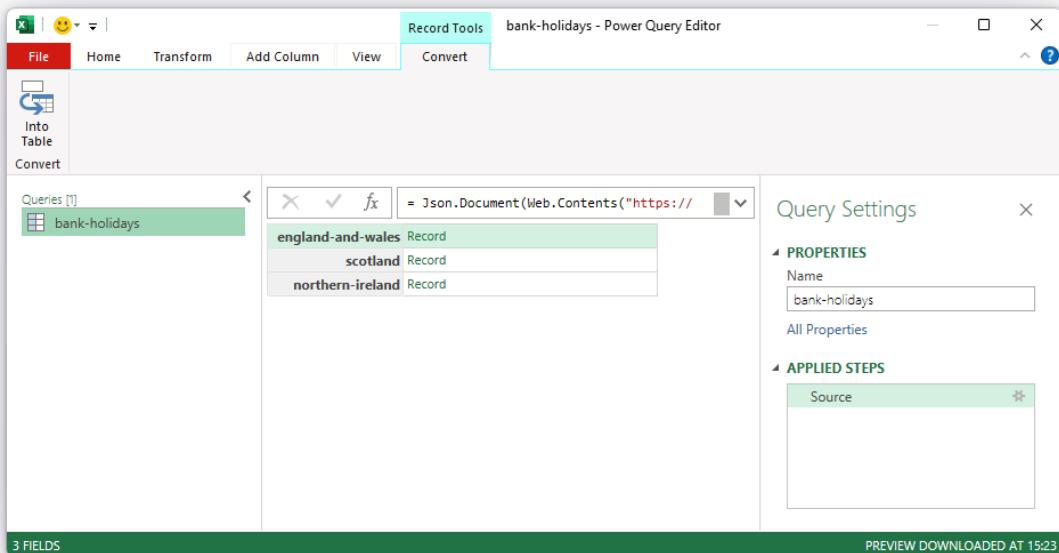


Clicking a cell reveals the cell contents in a panel at the bottom of the screen. This is useful for inspecting cells that contain a large amount of text.



## Important Idea or Concept

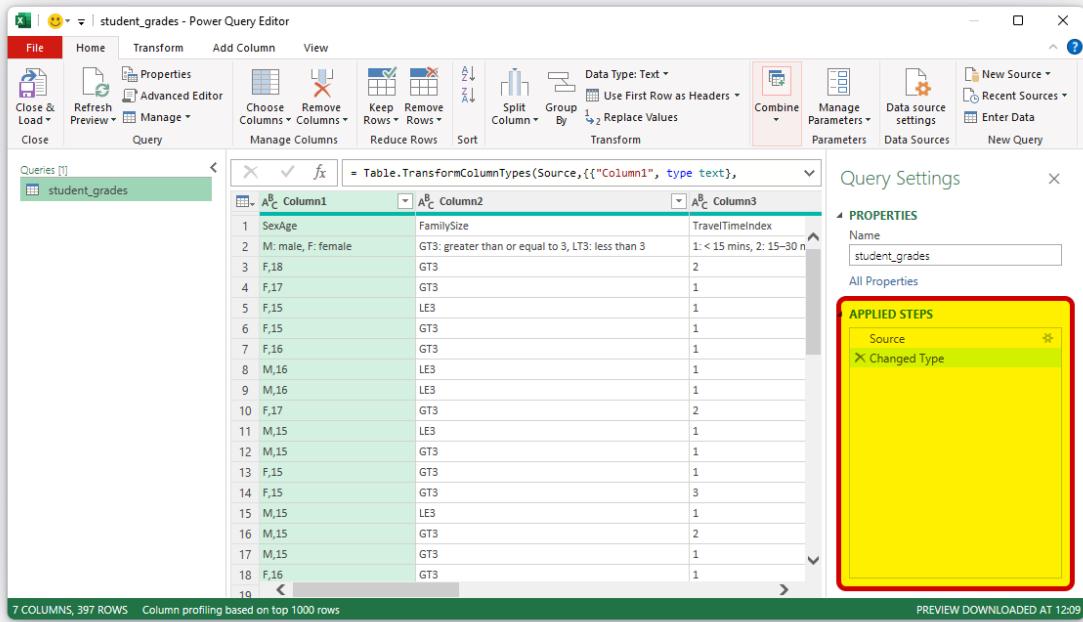
The data preview area displays tabular data as a grid of cells. However, it cannot display non-tabular data, such as a dictionary or a series of nested tables. Such non-tabular data is “tabularised” by representing a list with the word “List” and a dictionary with the word “Record”. These words are **clickable hyperlinks** that navigate into the collection.



**Figure 12:** The data preview area uses the word “Record” to represent a dictionary.

## 4.4 Applied Steps

The **Applied Steps** list appears at the bottom of the **Query Settings** panel. Each Step represents an operation applied to the dataset.



**Figure 13:** The Applied Steps list.

Examples of Steps include:

- Delete blank rows.
- Delete or hide a column.
- Convert column type to whole number, decimal number, text, date, etc.
- Split or merge columns.
- Create a new column by applying a formula.
- Merge or Append Queries.
- Group and aggregate.



### Helpful Hint

If you don't see the **Query Settings** panel, reopen it by navigating to **Ribbon > View > Query Settings**.



## Creating a Step

The buttons on the **Ribbon** are the easiest way to create Steps. Certain other actions will also create a Step, such as expanding a List or Record.



### Activity

Take a few minutes to browse the Ribbon and get an idea of the range of data transformations available.



### Activity

If you have imported “student\_grades.csv” into Power Query, try applying a few data transformations now. Notice that each transformation adds a Step to the bottom of the Applied Steps.

Don’t worry: all Steps can be undone!

## Navigating Steps

The Applied Steps can be navigated like an undo history. Clicking on a Step “rewinds” or “fast-forwards” to that Step.



### Key Point

The **last** Step of a Query is the **output**. Even if a different Step happens to be selected, the last Step is still the output.



## Key Point

A new Step is always inserted into the Applied Steps list immediately after the **currently selected Step**.  
The currently selected Step is highlighted in green.

## Deleting a Step

A Step can be deleted by hovering the mouse over the Step and clicking the **X** symbol.

The screenshot shows the Power Query Editor interface with the 'student\_grades' query loaded. The 'Applied Steps' list on the right side contains one step: 'Removed Columns'. This step is highlighted with a red circle and an X over it, indicating it is the currently selected step for deletion. The main area shows the query code: `= Table.RemoveColumns(#"Changed Type", {"Column1"})`. The preview pane at the bottom shows 6 COLUMNS, 397 ROWS.

**Figure 14:** Delete a step by clicking the **X** symbol.

Any Step except the first Step can be deleted this way.



## Activity

If you added any extra Steps to your **student\_grades** Query, delete them now. You will be left with just the two steps that you started with: “Source” and “Changed Type”.

## Editing a Step

Steps with a cog/gear symbol next to them are **editable**. A step can be edited even if it is not the final Step.



## Warning

Take care when inserting, deleting, or editing Steps, as it is possible to break the Query. For example, if you delete a column that is needed in a later Step, the Query will throw an error. If this happens, some repair work will be required!

## 4.5 Summary

- The **Queries** panel displays a list of all Queries in this workbook. Right-clicking a Query brings up a menu of useful commands.
- The **data preview** area displays a preview of the Query at the currently selected Step. Clicking on a cell reveals the full cell contents in a panel at the bottom of the screen.
- The **Applied Steps** list is a navigable history of all transformations applied in this Query. The final Step is the output of the Query.



## 5 Basic transformations

In this section, we will apply a series of basic transformations to our **student\_grades** Query.

If you would like to follow along, simply import “student\_grades.csv” into Power Query, as we did earlier. This creates a new Query called **student\_grades**. If you have applied further transformations to your version of **student\_grades**, delete them before you continue.

The screenshot shows the Microsoft Power Query Editor interface. The title bar says "student\_grades - Power Query Editor". The ribbon has tabs for File, Home, Transform, Add Column, and View. The Home tab is selected. The ribbon also includes Close & Load, Refresh, Advanced Editor, Properties, Manage Columns, Remove Columns, Keep Rows, Remove Rows, Sort, Split Column, Group By, Replace Values, Data Type: Text, Use First Row as Headers, Combine, Manage Parameters, Data source settings, Data Sources, New Source, Recent Sources, Enter Data, and New Query. The main area shows a table with three columns: Column1, Column2, and Column3. The first row contains column names: SexAge, FamilySize, and TravelTimeIndex. Subsequent rows show data points such as "M: male, F: female", "GT3: greater than or equal to 3, LT3: less than 3", and numerical values like 1, 2, 1, etc. The bottom status bar indicates "7 COLUMNS, 397 ROWS" and "Column profiling based on top 1000 rows". On the right side, there is a "Query Settings" pane with sections for Properties (Name: student\_grades) and Applied Steps (Source, Changed Type). The status bar at the bottom right says "PREVIEW DOWNLOADED AT 12:09".

**Figure 15:** The Power Query Editor window after importing “student\_grades.csv”. This is the starting point for this section.

You may wish to spend a few moments browsing the dataset before you continue. As an analyst, what would you do to clean up this dataset?

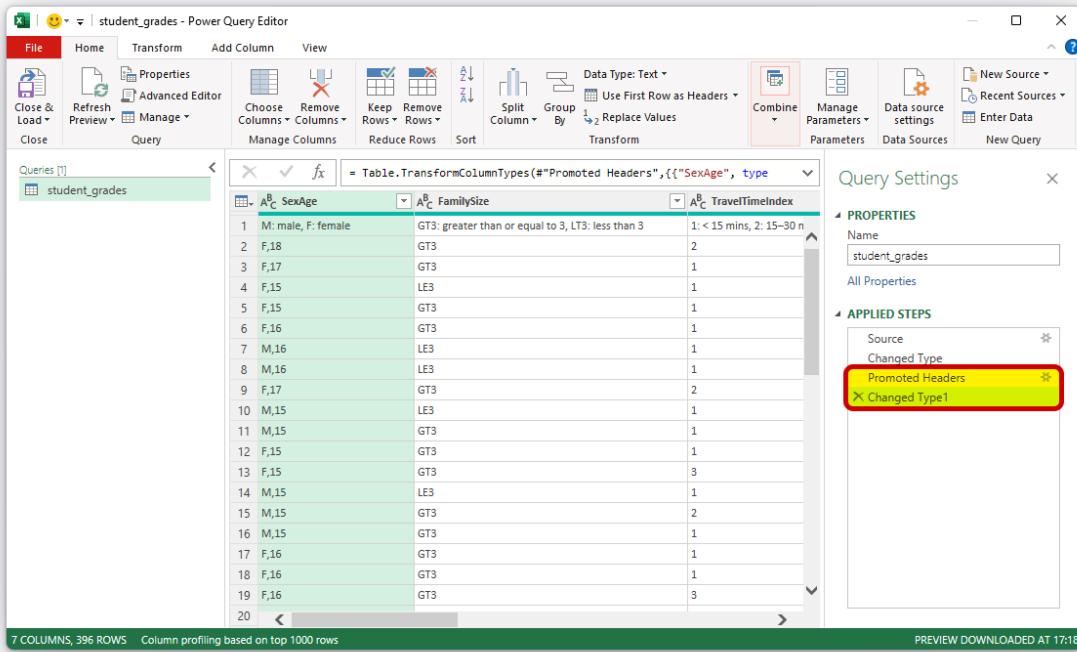
### 5.1 Use first row as headers

The column names are currently “Column1”, “Column2”, “Column3”, and so on. However, we can clearly see that the **first row** contains the true column names: “SexAge”, “FamilySize”, “TravelTimeIndex”, etc.

A <sup>B</sup> <sub>C</sub> Column1	A <sup>B</sup> <sub>C</sub> Column2	A <sup>B</sup> <sub>C</sub> Column3
1 SexAge	FamilySize	TravelTimeIndex
2 M: male, F: female	GT3: greater than or equal to 3, LT3: less than 3	1: < 15 mins, 2: 15–30 mins, 3: 30–60 mins, 4: > 60 mins
3 F,18	GT3	2
4 F,17	GT3	1
5 F,15	LE3	1
6 F,15	GT3	1
7 F,16	GT3	1
8 M,16	LE3	1
9 M,16	LE3	1
10 F,17	GT3	2
11 M,15	LE3	1
	GT3	

**Figure 16:** The row containing the column names incorrectly appears as the first data row.

1. To use the first row as the column headers (column names), navigate to **Ribbon > Home > Transform group > Use First Row as Headers > Use First Row as Headers**.
2. The column names are now correctly in the header row. The number of data rows has changed from 397 to 396.
3. Notice that Power Query appends not one, but two Steps to the Applied Steps list. We get a “Changed Type” Step for free with certain transformations. Here, “Changed Type1” is automatically converting each column to what Power Query thinks is the most appropriate type.



**Figure 17:** The **Use First Row as Headers** command creates a Step called “Promoted Headers”.

4. We actually don’t need “Changed Type” or “Changed Type1”. Let’s tidy up by **deleting** both Steps. This leaves just “Source” and “Promoted Headers”.

## 5.2 Remove the secondary header row

We notice that **student\_grades** has two header rows instead of the typical single row. The second row clearly contains metadata that describes each column. This is useful information; however, it could interfere with our analysis. For example, we cannot take the average of a numeric column if one of the values is text.

1. Navigate to **Ribbon > Home**. Our goal is to **remove the first row**. Find a command on this tab that will achieve this. Execute the command to add it as a Step.
2. Notice that the number of rows has gone from 396 to 395.



Solution (select text to copy):

## 5.3 Add an index column

1. The table lacks a unique row ID. Let's add one by navigating to **Ribbon** > **Add Column** > **General** group > **Index Column**. In this case, it doesn't matter whether we then choose **From 0** or **From 1**. Let's choose **From 1**.

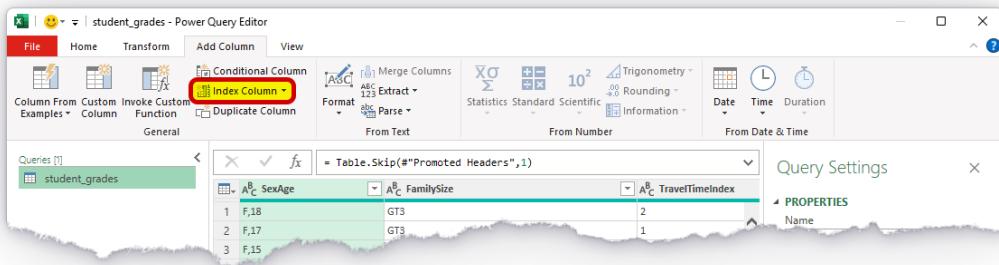


Figure 18: The **Index Column** command is found on the **Add Column** tab of the Ribbon.

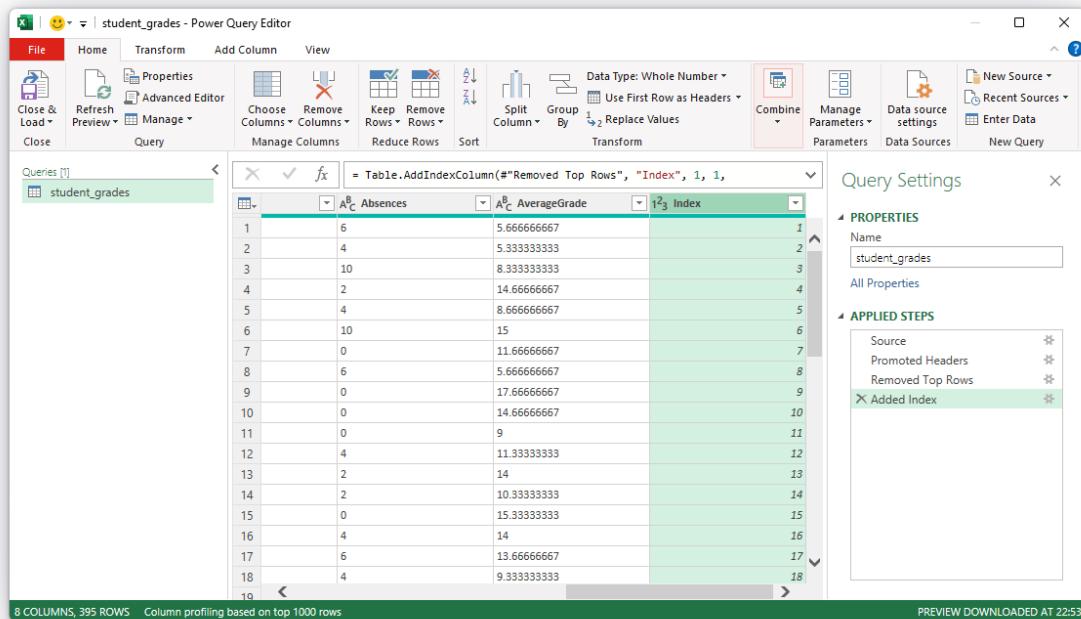


Figure 19: The new index column is inserted as the last column.

2. In the bottom-left corner, we can see that we now have 8 columns.
3. The index column has been inserted as the last column. It is more usual for an index column to be the first column. To move the column “Index” to the first position, **right-click** the column > **Move** > **To Beginning**.

## 5.4 Change column types

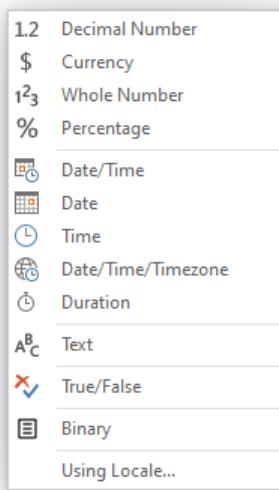
---

The small symbol to the left of each column header indicates the column **type**.

	Index	SexAge	FamilySize	TravelTimeIndex	StudyTimeIndex
1	1	F,18	GT3	2	2
2	2	F,17	GT3	1	2
3	3	F,15	LE3	1	2
4	4	F,15	GT3	1	3
5	5	F,16	GT3	1	2
6	6	M,16	LE3	1	2
7	7	M,16	LE3	1	2
8	8	F,17	GT3	2	2
9	9	M,15	LE3	1	2
10	10	M,15	GT3	1	2
11	11	F,15	GT3	1	2
12	12	F,15	GT3	1	2

**Figure 20:** The column type is indicated to the left of each column header.

The available column types are shown in Figure 21.



**Figure 21:** The available column types in Power Query.

In **student\_grades**, notice that the “Index” column has the symbol “ $1^2_3$ ” – Whole Number, while all the other columns have the symbol “ $A^B_C$ ” – Text.



### Important Idea or Concept

Column types are an important concept in Power Query. Different column types enable different transformations. For example, mathematical calculations can be applied to numeric columns, and calendar calculations can be applied to date columns.



### Warning

If you try to convert a column to an unsuitable type, the cells that cannot be converted to that type will become **Error** cells.



There are several ways of converting column types.

### Method 1: manually, one column at a time

1. Identify one of the columns of **student\_grades** that you think should be converted to a different type.
2. Click the **type** symbol to the left of the column header (e.g., “ $A^B_C$ ”).
3. Choose a new column type from the dropdown list.



#### Activity

Try Method 1 on one column of **student\_grades**.

### Method 2: manually, any number of columns at a time

1. **Ctrl + click** to select multiple columns that you would like to convert to the same type.
2. Navigate to **Ribbon > Transform**. Find a command on this tab that will convert the selected columns to the desired type. Execute this command.



#### Activity

Try Method 2 on two other columns of **student\_grades**.



Solution (select text to copy):



### Helpful Hint

**Ctrl + click** on column headers to select multiple columns. **Shift + click** to select a contiguous range of columns. **Ctrl + A** to select all columns.

### Method 3: automatically, any number of columns at a time

Power Query can attempt to automatically detect the most suitable column type.

1. Select the columns that you would like Power Query to convert to what it thinks is the most suitable type. If you are not sure which columns to select, press **Ctrl + A** to select all columns.
2. Navigate to **Ribbon > Transform**. Find a command on this tab that will perform automatic type conversion. Execute this command.
3. Browse the data preview area and make sure that each column was successfully converted to an appropriate type.

A <sup>B</sup> C	Romantic	1 <sup>2</sup> 3	Absences	1.2	AverageGrade
no			6	5.666666667	
no			4	5.333333333	
no			10	8.333333333	
yes			2	14.666666667	
no			4	8.666666667	
no			10	15	
no			0	11.666666667	
no			6	5.666666667	
no			0		

**Figure 22:** Each column of **student\_grades** has been converted to an appropriate type.

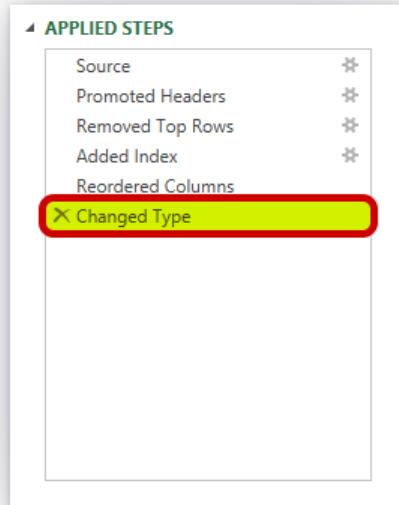


## Activity

Try Method 3 on all remaining columns of **student\_grades**.

Solution (select text to copy):

Notice that we have only a single “Changed Type” Step in our Applied Steps, even though we have changed the types of multiple columns. Power Query often condenses consecutive Steps of the same kind into a single Step.



**Figure 23:** Power Query has condensed our “Changed Type” Steps into a single Step.

## 5.5 Remove empty rows

In the **data preview** area of Query **student\_grades**, scroll down to the final row, 395. The data preview features “lazy loading”: it loads just enough rows to fill the screen, but will load more as we scroll down. This saves resources, especially for very large tables.



## Key Point

To scroll faster, use the **Page Up** and **Page Down** keys.

**Figure 24:** The final rows of the **student\_grades** Query.

We can see 5 empty rows at the bottom of the table. Let's try to remove these using the **Remove Blank Rows** command.

1. Navigate to **Ribbon** > **Home**. Find and execute the **Remove Blank Rows** command.
2. A “Removed Blank Rows” Step appears in the Applied Steps. However, we still have 395 rows. **Why?**



## Activity

Why were no rows removed? How could we fix this?



Solution (select text to copy):

Now that you know why the **Remove Blank Rows** command had no effect, the next step is to fix the problem.



### Activity

Think of what Steps you could apply to delete the almost-empty rows. There are several approaches you might take. Execute your plan.

When you are done, your **student\_grades** Query should have **390** rows.

Hopefully, your plan worked and you now have 390 rows! If not, try one of the four approaches below.

#### Approach 1: backtrack the last few Applied Steps

We understand that the reason the **Remove Blank Rows** command had no effect is due to the “Index” column, which is not blank.

1. Delete the last few Applied Steps, including “Added Index”.
2. Rebuild the deleted Steps, excluding “Added Index” and “Reordered Columns”.



### Key Point

Notice that the small green bar below each column header now contains a small amount of grey. Grey indicates empty values. Hover your mouse over the green/grey bar to view a **Column Quality** popup.

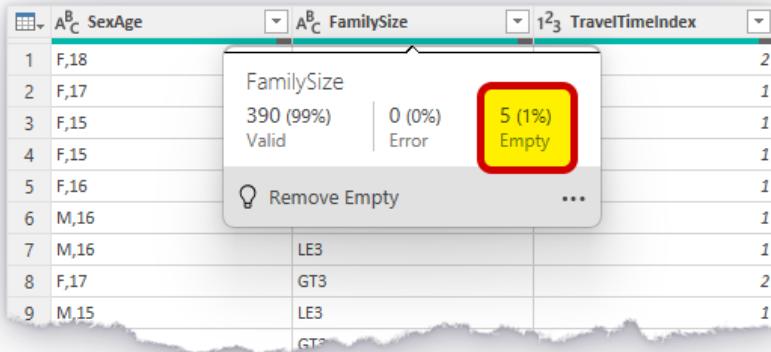


Figure 25: The **Column Quality** popup.

3. Execute **Remove Blank Rows**. The 5 empty rows are successfully deleted.
4. Add a new index column.



### Warning

If we delete *only* the “Added Index” Step, this breaks the Query, since some of the later steps are trying to transform a column called “Added Index”.



### Useful Tool

Broken Queries can often be recovered by editing the raw code that comprises the Applied Steps. This code is called **M language** and can be edited in the **Advanced Editor**, accessible via the Ribbon. We will not cover the Advanced Editor in this apprenticeship; however, you may wish to check it out if you are interested.

## Approach 2: delete and recreate the index column

This approach is very similar to Approach 1, but less tidy.

1. Delete the “Index” column by **right-clicking** the column > **Remove**.
2. Execute **Remove Blank Rows**.
3. Add a new index column.

## Approach 3: filter

You may have noticed the **filter** arrow next to each column header.

Index	SexAge	FamilySize	TravelTimeIndex	StudyTimeIndex
1	F,18	GT3	2	2
2	F,17	GT3	1	2
3	F,15	LE3	1	2
4	F,15	GT3	1	3
5	F,16	GT3	1	2
6	M,16	LE3	1	2
7	M,16	LE3	1	2
8	F,17	GT3	2	2
9	M,15	LE3	1	2
10	M,15	GT3	1	2
11	F,15	GT3	1	2
12	F,15	GT3		

**Figure 26:** Use the filter arrow to filter and sort a column.

1. Filter any column to exclude the value “(blank)”.

## Approach 4: delete bottom rows

We notice that the empty rows are the bottom 5 rows.

1. Navigate to **Ribbon** > **Home** tab > **Remove Rows** > **Remove Bottom Rows**.
2. Enter value **5** to remove 5 rows.

## 5.6 Arithmetic

The “AverageGrade” column of **student\_grades** has a lot of values that end in either ...3333 or ...6667.

1.2 AverageGrade
5.6666666667
5.3333333333
8.3333333333
14.6666666667
8.6666666667
15
11.6666666667
5.6666666667

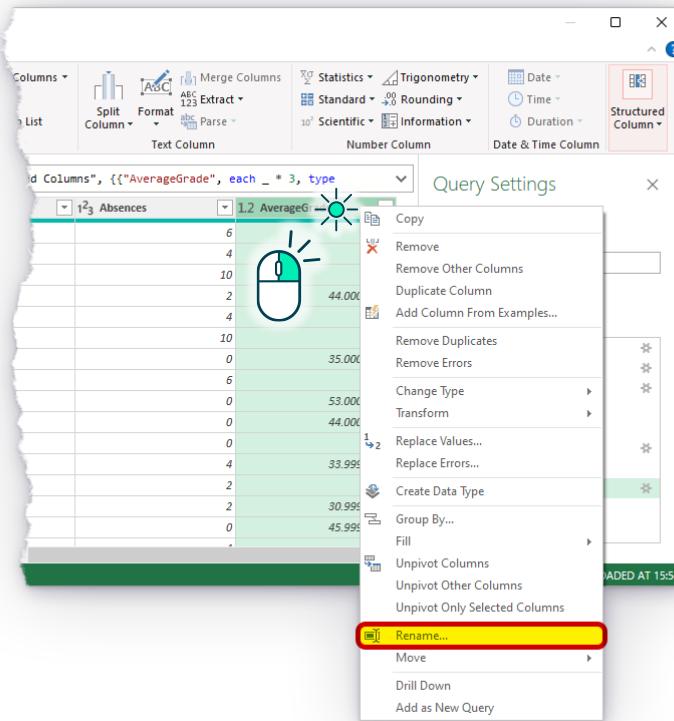
**Figure 27:** The values in the “AverageGrade” contain many digits after the decimal point.

As you will have guessed, “AverageGrade” is the average of 3 grades. Let’s convert this to the total grade by multiplying by 3.

1. Click the “AverageGrade” header to select that column.
2. Navigate to **Ribbon > Transform > Number Column** group. Find a command in this group called “Multiply”. Execute the command. Multiply by the value **3**.

Solution (select text to copy):

3. **Rename** this column “OverallGrade”.



1.2 OverallGrade
17
16
25
44.00000001
26
45
35.00000001
17
53.00000001

**Figure 28:** The “OverallGrade” column.

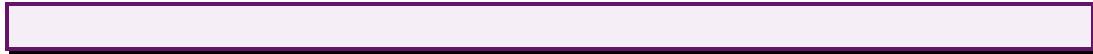
## 5.7 Rounding

The “OverallGrade” column looks a bit messy. We can see that many values end in ...0001 or ...9999. This is a **rounding error** and is an artefact of binary-to-decimal conversion.

Let's fix the problem by rounding the numbers.

1. Navigate to **Ribbon > Transform > Number Column** group Find a command in this group that will round to the **nearest** value (note: **not** round up/down). Choose a sensible number of decimal places and execute the command.

Solution (select text to copy):



1.2 OverallGrade
17
16
25
44.00000001
26
35.00000001
17
53.00000001
44.00000001
27
33.99999999

1.2 OverallGrade
17
16
25
44
26
45
35
17
53
44
27
34
12

**Figure 29:** After rounding, “OverallGrade” should consist of integer numbers only.

2. The column type of “OverallGrade” is currently “1.2” – Decimal Number. Change it to a more suitable type.

## 5.8 Split a column

In the **student\_grades** Query, we can see that the “SexAge” column contains both the sex and age of the student. Let’s split this column into two columns.

1. Select the “SexAge” column.
2. Find the **Split Column** command. This is a common command and is located on both the **Home** and **Transform** tabs of the Ribbon.
3. Inspect the **Split Column** options: “By Delimiter”, “By Number of Characters”, etc.



## Activity

Take a few minutes to run some experiments with the various **Split Column** options.

When you are finished, remember to **delete** the Steps.

4. **Split** the “SexAge” column into two columns using any **Split Column** method you like.
5. **Rename** the columns “Sex” and “Age”.

The screenshot shows the Power Query Editor interface with the 'student\_grades' query selected. The 'Transform' tab is active. A context menu is open over the 'SexAge.1' column, listing options like Copy, Remove, Rename..., and others. The 'Rename...' option is highlighted with a red box. The 'Query Settings' pane on the right shows the query name is 'student\_grades' and lists applied steps including 'Changed Type1'.

**Figure 30:** Rename a column.

A <sup>B</sup> <sub>C</sub>	Sex	t <sup>2</sup> <sub>3</sub>	Age
	F		18
	F		17
	F		15
	F		15
	F		16
	M		16
	M		16
	F		17
	M		15
	M		

**Figure 31:** The completed “Sex” and “Age” columns.

Notice that Power Query automatically applied a “Changed Type” Step, which converted the “Age” column to type “t<sup>2</sup><sub>3</sub>” – Whole Number.

## 5.9 Find and replace

---

The “Sex” column currently contains two values: “F” and “M”. Let’s change these to “Female” and “Male”.

1. Select the “Sex” column.
2. Navigate to **Ribbon > Transform** tab. Find a command on this tab that can replace values. Execute the two text replacements.

Solution (select text to copy):

A	B	C	Sex
			F
			F
			F
			F
			F
			M
			M
			F
			M
			M
			F
			F

A	B	C	Sex
			Female
			Male
			Male
			Female
			Male
			Male
			Female
			Female

**Figure 32:** Before and after the “Sex” change.

## 5.10 Summary

- The **Home** tab of the Ribbon contains the **Close & Load** button and various popular commands.
- The **Transform** tab of the Ribbon contains commands for transforming data, splitting and merging columns, reshaping the table, and applying aggregate functions.
- The Add Column tab of the ribbon contains the popular commands **Column From Examples** and **Custom Column**, as well as transformation commands that duplicate the original column.

## 6 Merging Queries

Take a look at the “TravelTimeIndex” and “StudyTimeIndex” columns. What’s wrong?

1 <sup>A</sup> <sub>B</sub> TravelTimeIndex	1 <sup>A</sup> <sub>B</sub> StudyTimeIndex
2	2
1	2
1	2
1	3
1	2
1	2
1	2
2	2
1	2

**Figure 33:** Could we replace these columns with versions that provide more insightful data?

In step 5.2, we deleted the secondary header row. That row contained metadata about the columns. We can review that metadata now.

A <sup>B</sup> <sub>C</sub> TravelTimeIndex	A <sup>B</sup> <sub>C</sub> StudyTimeIndex
1: < 15 mins, 2: 15–30 mins, 3...	Weekly study time. 1: < 2 hrs...
2	2
1	2
1	2
1	3
1	2
1	2
1	2
2	2

**Figure 34:** The secondary header row, which contained metadata.



## Activity

In your **Applied Steps** list, click the first Step, “Source”, and inspect the metadata for the “TravelTimeIndex” and “StudyTimeIndex” columns.



“TravelTimeIndex” column:

1: < 15 mins, 2: 15–30 mins, 3: 30–60 mins, 4: > 60 mins

“StudyTimeIndex” column:

Weekly study time. 1: < 2 hrs, 2: 2–5 hrs, 3: 5–10 hrs, 4: > 10 hrs

For each of these columns, we would like to replace the index number with the value. For example, for “TravelTimeIndex”, we would like to replace 1 with “< 15 mins”, 2 with “15–30 mins”, and so on.

There are several ways to achieve this in Power Query. We have already seen **Replace Values**. This time, we will use the more versatile method of creating a **lookup table**, which we will then **merge** (join) to the **student\_grades** Query.

## 6.1 Create lookup tables

We will create our lookup tables in Excel. To close Power Query and get back to Excel, find the **Close & Load** command.



**Figure 35:** The **Close & Load** command in Power Query is located on the Home tab of the Ribbon.

1. Create a new worksheet and name it “Lookup tables”.
2. On the **Lookup tables** worksheet, manually populate a  $5 \times 2$  range of cells with the “TravelTimeIndex” lookup values, as below.

	A	B	C
1	TravelTimeIndex	TravelTime	
2		1 < 15 mins	
3		2 15–30 mins	
4		3 30–60 mins	
5		4 > 60 mins	
6			
7			

**Figure 36:** Preparing a “TravelTime” lookup table.

3. Convert this range to an **Excel Table**: select the cells and press **Ctrl + T**.

A	B	C	D	E	F
1	TravelTimeIndex	TravelTime			
2		1 < 15 mins			
3		2 15–30 mins			
4		3 30–60 mins			
5		4 > 60 mins			
6					
7					
8					
9					
10					

**Figure 37:** Convert range to Table.

	A	B	C	D
1	TravelTimeIndex	TravelTime		
2		1 < 15 mins		
3		2 15–30 mins		
4		3 30–60 mins		
5		4 > 60 mins		
6				
7				
8				

**Figure 38:** The completed **tblTravelTime**.



### Important Idea or Concept

**Tables** are powerful data containers within Excel, and they carry several advantages over ordinary ranges of cells. One useful feature is that a **reference** to a Table dynamically reflects the current size of the Table. This means that if we change a Table by adding or removing rows or columns, references will still “see” the full Table.

We will use Excel Tables frequently in this apprenticeship.

4. Rename the Table “tblTravelTime”. To do this, navigate to **Ribbon** > **Table Design** tab > **Properties** group > **Table Name**: Type the table name and press **Enter**.

A	B
1	TravelTimeIndex
2	1 < 15 mins
3	2 15–30 mins
4	3 30–60 mins
5	4 > 60 mins
6	
7	

**Figure 39:** How to rename a Table.

5. Repeat steps 2–4 to create a second lookup table, “tblStudyTime”. You can place this on the same sheet as **tblTravelTime**.

A	B
1	TravelTimeIndex
2	1 < 15 mins
3	2 15–30 mins
4	3 30–60 mins
5	4 > 60 mins
6	
7	
8	StudyTimeIndex
9	1 < 2 hrs
10	2 2–5 hrs
11	3 5–10 hrs
12	4 > 10 hrs
13	
14	

**Figure 40:** The completed **tblTravelTime** and **tblStudyTime**.

## 6.2 Import lookup tables into Power Query

We will first import **tblTravelTime**.



1. Navigate to **Ribbon** > **Data** tab > **Get & Transform Data** group > **Get Data**. Our goal is to **import a Table into Power Query**. Find a command on this menu that will achieve this. Do not execute the command just yet.

Solution (select text to copy):



## Important Idea or Concept

When we click **Get Data** > **From Other Sources** > **From Table/Range**, the behaviour is different depending on what types of cells are currently selected.

Firstly, the selection must be a **contiguous rectangle**.

If the selection touches any part of a **Table**, Power Query will use the entire Table as the source range.

If the selection touches any part of a **Named Range**, Power Query will use the entire Named Range as the source range.

If the cell selection touches ordinary cells only, Power Query will **convert** those cells into a Table and use that Table as the source range.

2. Being mindful of the information above, import **tblTravelTime** into Power Query.
3. Rename this Query “qryTravelTime”. This will help us distinguish **tblTravelTime** from **qryTravelTime**. Things could get confusing otherwise!



The screenshot shows the Power Query Editor interface. The 'Queries' panel on the left lists 'student\_grades' and 'qryTravelTime'. The main area displays a table with four rows and two columns. The first column is labeled 'TravelTimeIndex' and the second is 'TravelTime'. The data is as follows:

TravelTimeIndex	TravelTime
1	< 15 mins
2	15-30 mins
3	30-60 mins
4	> 60 mins

The 'Applied Steps' pane on the right shows a single step named 'Changed Type'.

**Figure 41:** Our “TravelTime” lookup table has been imported as a Query.

To import **tblStudyTime**, we have two options. Method 1 is easier; Method 2 is fancier. Choose whichever you like.

## Method 1: import from Excel

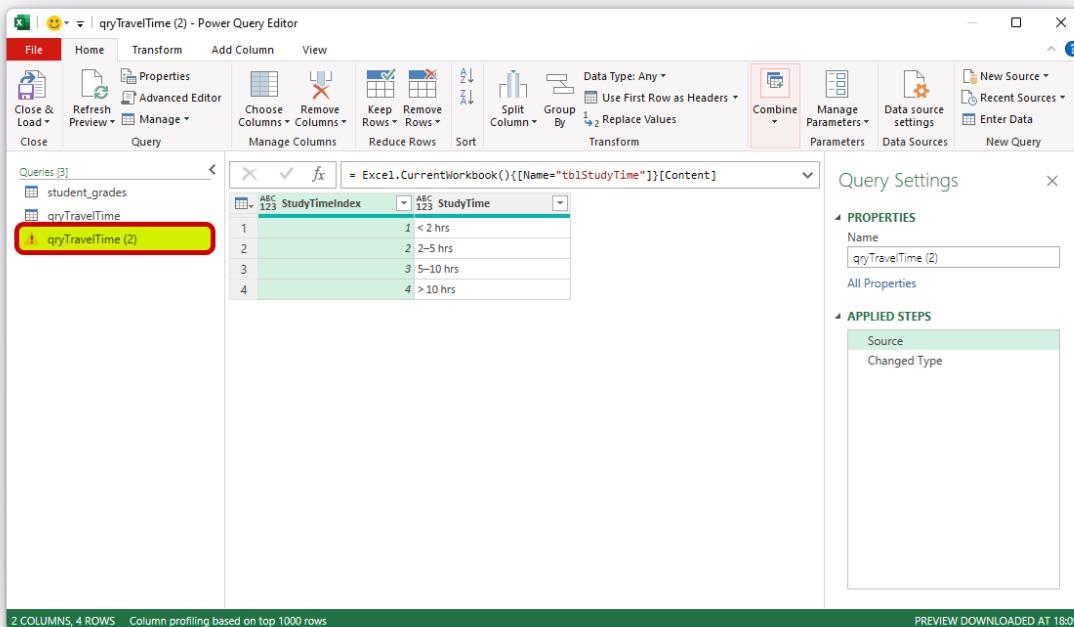
1. **Close & Load** to get back to Excel.
2. Import **tblStudyTime** the same way you imported **tblTravelTime**.

## Method 2: duplicate and edit the **tblTravelTime** Query

1. In Power Query, navigate to the **Queries** panel > **right-click** the **qryTravelTime** Query > **Duplicate**. This creates an independent clone of the Query.
2. In the **Applied Steps** list, click **Source**.
3. Above the **data preview** area, find the **Formula Bar**.

```
= Excel.CurrentWorkbook(){[Name="tblTravelTime"]}[Content]
```

4. This is a snippet of M language that imports **tblTravelTime**. We do not need to know anything about M language to guess that we could probably change the text `Name="tblTravelTime"` to `Name="tblStudyTime"`. Let's see if it works! Make the edit and press **Enter**.



**Figure 42:** After editing the M formula to replace “tblTravelTime” with “tblStudyTime”. The source has updated correctly; however, there is now an error in the Query.

5. If all went well, you should see the Query update to reflect the **tblStudyTime** Table, as in Figure 42.



### Helpful Hint

If you instead got “Expression.Error: We couldn't find an Excel table named 'tblStudyTime'”, go back to Excel and fix the Table name.

6. We still have a problem to fix. Note the warning symbol in the **Queries** panel next to **qryTravelTime (2)**. Diagnose and fix this problem.



Solution (select text to copy):



7. Rename this Query “qryStudyTime”.

The screenshot shows the Power Query Editor interface. The ribbon tabs are Home, Transform, Add Column, and View. The Home tab is selected. The Queries panel on the left shows three queries: student\_grades, qryTravelTime, and qryStudyTime, with qryStudyTime selected. The main preview pane displays a table with two columns: StudyTimeIndex (1, 2, 3, 4) and StudyTime (< 2 hrs, 2-5 hrs, 5-10 hrs, > 10 hrs). The Transform ribbon group is visible, showing various data type conversion and manipulation tools. The Query Settings pane on the right shows the Name is set to qryStudyTime and the Applied Steps list contains a single step named "Changed Type".

**Figure 43:** The completed **qryStudyTime**.

## 6.3 Merge lookup tables

In this section, our goal is to modify the **student\_grades** Query so that the “TravelTimeIndex” and “StudyTimeIndex” columns are replaced with the associated “TravelTime” and “StudyTime” columns from our lookup tables.

We will achieve this by **merging** (joining) **qryTravelTime** and **qryStudyTime** to **student\_grades**.

Let's start with **qryTravelTime**.

1. In the **Queries** panel, select the **student\_grades** Query.
2. Navigate to **Ribbon** > **Home** tab > **Combine** group > **Merge Queries** > **Merge Queries**.

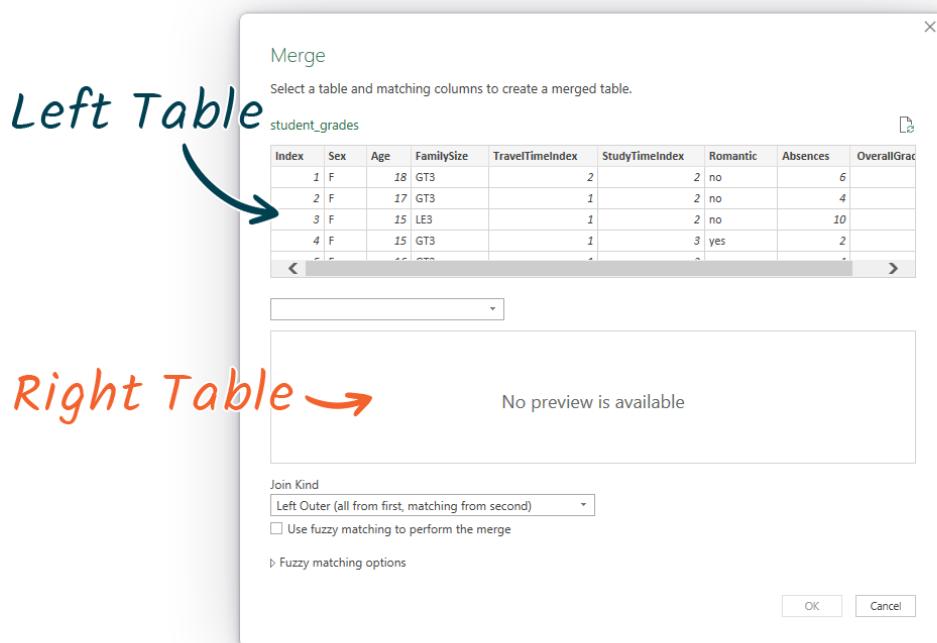


## Definition

The other option, **Merge Queries as New**, would create a **new** Query whose Source is a Merge command. This is useful, but not what we want in this instance. We want **Merge Queries**.

3. The **Merge** window appears. We need to provide the following parameters:

- The Right Table. (Note: the Left Table is **student\_grades**.)
- What column(s) to join on.
- The type of join.

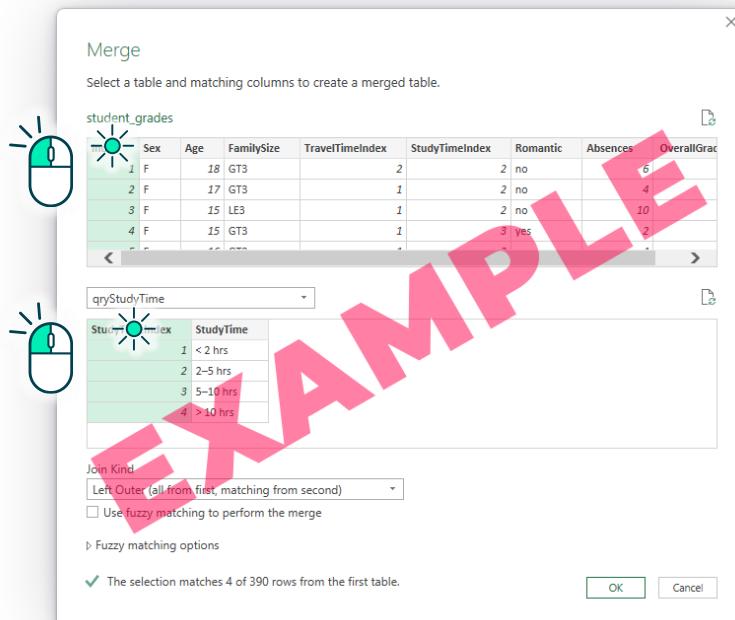


**Figure 44:** The Merge window. In Power Query, a “merge” is the equivalent of a join.



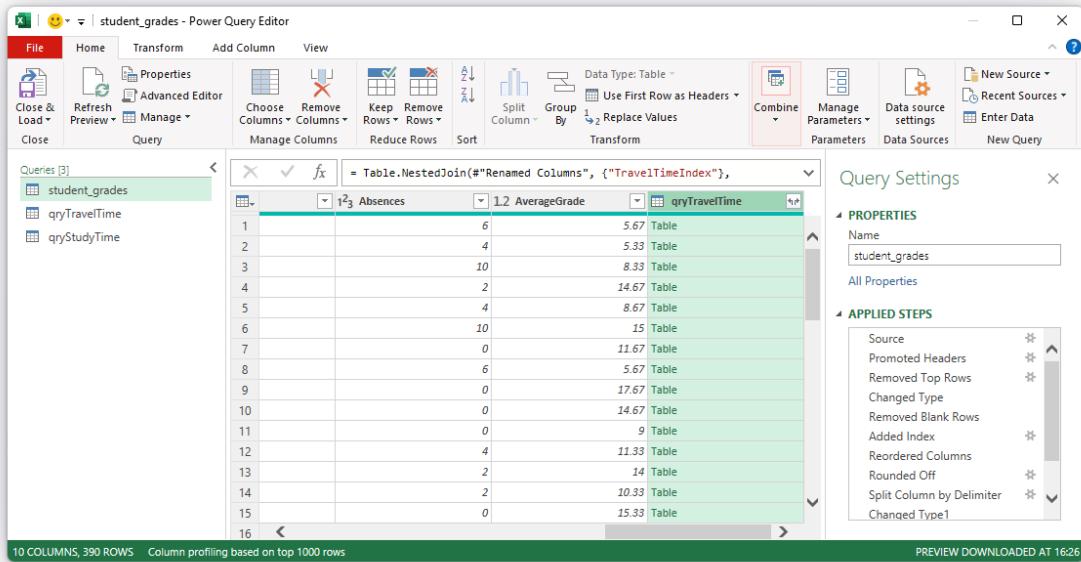
## Helpful Hint

The way we must specify the join columns is not entirely intuitive. In the **Merge** window, click a column header of the Left Table to select it. Click the corresponding column header of the Right Table. This pair of columns will now be matched. This is illustrated in **Figure 45**.



**Figure 45:** How to select the join columns in the Merge window. Note: in the screenshot, the tables and columns were selected arbitrarily for example only.

4. Use your best judgement to input suitable parameters, then click **OK**.
5. The **student\_grades** Query now has an extra column: “qryTravelTime”. Notice that every row contains the same value: **Table**. The green text indicates a clickable hyperlink.



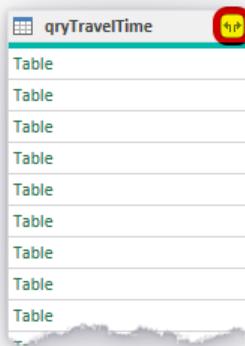
**Figure 46:** After merging **student\_grades** to **qryTravelTime**.

### Important Idea or Concept

In Power Query, green text indicates a **collection** of values. This collection can be a list, a dictionary, or a table. We can navigate into one of these collections by clicking the green hyperlink text. We can undo the navigation by deleting the corresponding Applied Step.

You should experiment to familiarise yourself with this behaviour.

6. Power Query allows us to expand all the **Table** collections at the same time. To the right of the “qryTravelTime” column header, click the ↗ symbol.



**Figure 47:** Click the symbol to expand a column after a Merge operation.

7. In the dropdown box, check the columns you would like to take from **qryTravelTime**. (Generally, we prefer not to bring in any redundant columns.)
8. Either check or uncheck “Use original column name as prefix” as you see fit. Run some experiments to see the effect. Click **OK**.
9. You should now have a new column: “TravelTime”. You may optionally move this column so that it sits next to “TravelTimeIndex”. This can be achieved by dragging the column header.

1	2	3	TravelTimeIndex	A	B	C	TravelTime
				2	15–30 mins		
				1	< 15 mins		
				1	< 15 mins		
				1	< 15 mins		
				1	< 15 mins		
				1	< 15 mins		
				1	< 15 mins		
				2	15–30 mins		
				1	< 15 mins		
				1	< 15 mins		
				1	< 15 mins		
				3	30–60 mins		
				1	< 15 mins		
				2	> 30 mins		

**Figure 48:** After merging **student\_grades** to **qryTravelTime**, we can see that the correct “TravelTime” has been retrieved for each “TravelTimeIndex”.

10. With a brief visual check, verify that the merge was successful. If all looks good, then we no longer need the “TravelTimeIndex” column. Navigate to **Ribbon > Home** and find a command to delete the column.



### Helpful Hint

There are two different commands for deleting columns in Power Query. Both commands are found under **Ribbon > Home** tab > **Manage Columns** group. The difference is that the **Choose Columns** command is editable (⚙️) whereas the **Remove Columns** command is not.

When in doubt, use **Choose Columns**.

11. We have merged **student\_grades** to **qryTravelTime** and brought in the column “TravelTime”. Now use the same technique to bring in the column “StudyTime”.

1 <sup>2</sup> <sub>3</sub> TravelTimeIndex	A <sup>B</sup> <sub>C</sub> TravelTime	1 <sup>2</sup> <sub>3</sub> StudyTimeIndex	A <sup>B</sup> <sub>C</sub> StudyTime
2   15–30 mins			2   2–5 hrs
1   < 15 mins			2   2–5 hrs
1   < 15 mins			2   2–5 hrs
1   < 15 mins			2   2–5 hrs
1   < 15 mins			3   5–10 hrs
1   < 15 mins			2   2–5 hrs
1   < 15 mins			2   2–5 hrs
2   15–30 mins			2   2–5 hrs
1   < 15 mins			2   2–5 hrs
1   < 15 mins			2   2–5 hrs
3   30–60 mins			3   5–10 hrs
1   < 15 mins			1   < 2 hrs
2   15–30 mins			2   2–5 hrs
1   < 15 mins			3   5–10 hrs
1   < 15 mins			1   < 2 hrs
1   < 15 mins			

**Figure 49:** After merging **student\_grades** to both **qryTravelTime** and **qryStudyTime**. Once you have verified that the join was successful, you may delete the columns “TravelTimeIndex” and “StudyTimeIndex”.



## Definition

In SQL, the definition of a join specifies that NULL (an empty cell) **does not** match with another NULL.

In Power Query, the equivalent *null* or “” **does** match with another *null* or “”.

## 6.4 Summary

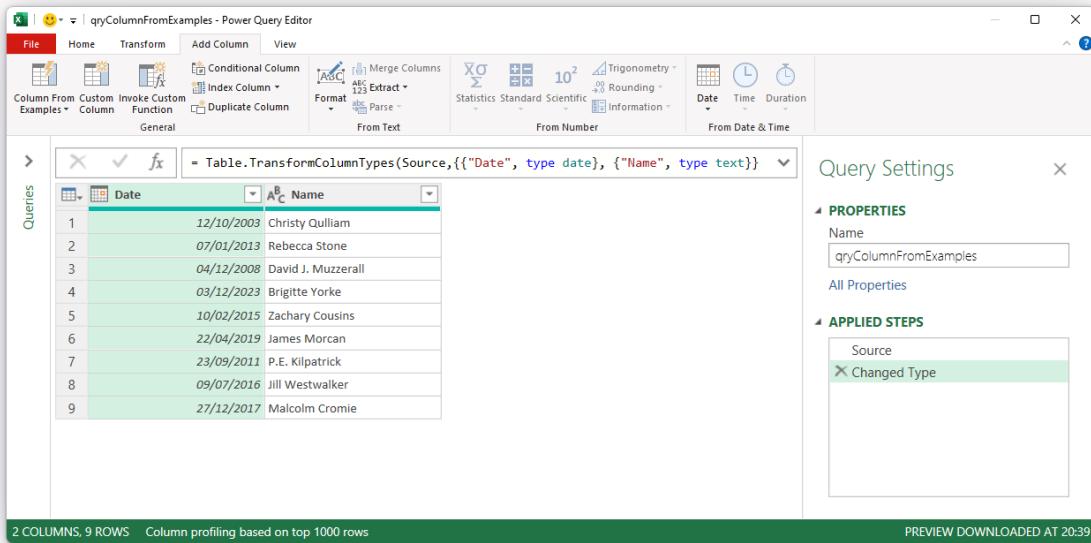
- An Excel Table can be imported into Power Query by selecting a cell in the Table and navigating to **Ribbon** > **Data** tab > **Get & Transform Data** group > **Get Data** > **From Other Sources** > **From Table/Range**.
- The Merge command performs a **join** between two tables.
- After a Merge, we must remember to click the symbol to choose which columns to bring in from the Right Table.

## 7 Column From Examples

**Column From Examples** is a powerful tool within Power Query for creating a formula-based column without requiring us to actually write any formulas.

With Column From Examples, we provide Power Query with examples of the output we desire, and Power Query will try to intelligently compose a formula that produces that output.

If you want to follow the worked examples in this section, find the provided file “column\_from\_examples.csv” and import this into Power Query. We will call this query “qryColumnFromExamples”.

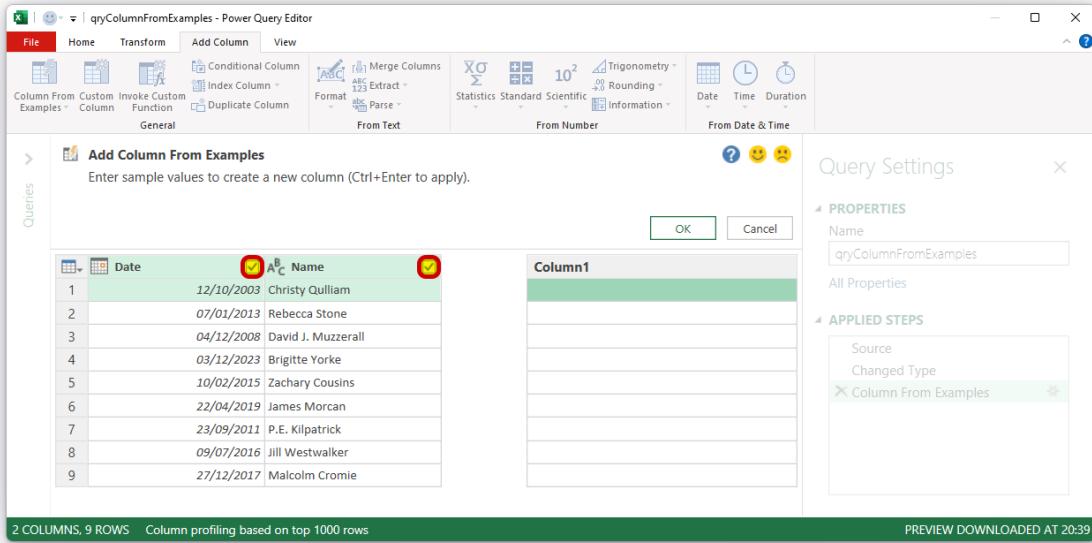


**Figure 50:** “column\_from\_examples.csv” imported into Power Query.

## 7.1 Extract weekday name

Let’s see if we can get Power Query to extract the **weekday name** from the “Date” column

1. If you are following the examples, select **qryColumnFromExamples** and navigate to **Ribbon > Add Column tab > General group > Column From Examples**.
2. In the dropdown menu, there are two options: **From All Columns** and **From Selection**. This indicates which columns may be used as “ingredients” in the generated formula. Let’s choose **From All Columns**.
3. The **Add Column From Examples** interface opens. Note the checkmark above each column. Since we selected **From All Columns** in the previous step, all columns are checked.



**Figure 51:** Add Column From Examples. The checkmarks indicate which columns Power Query will use in the formula that generates the new column.

4. We are trying to extract the weekday name from the “Date” column. Since we don’t need the “Name” column, we can **uncheck** it.
5. There is a space to the right, below “Column1”, where we can write our desired outputs. The first date is 12/10/2003, so our first output should be whatever the weekday was on that date. Fortunately, there is an easy way of finding this out. With the cell immediately below “Column1” selected, press **Shift + Space**. This brings up a list of **suggested outputs**.



The screenshot shows the Power Query Editor interface. In the center, there's a table with two columns: 'Date' and 'Name'. The 'Column1' dropdown menu is open, displaying a list of suggested outputs. The list includes 'Sunday' (Day of Week Name from Date), '12' (Day from Date), and many other date-related functions like '285' (Days of Year from Date) and '2003-10-12T23:59:59.999999' (End of Day from Date). The 'Properties' pane on the right shows the query name as 'qryColumnFromExamples'. The 'Applied Steps' pane shows the 'Column From Examples' step applied.

**Figure 52:** Pressing **Shift + Space** brings up a list of suggested outputs.

6. We notice “Sunday” is among the suggested outputs, so let’s type “Sunday” in the corresponding cell of “Column1” and see what happens.

The screenshot shows the Power Query Editor interface. The 'Column1' dropdown menu is open, and 'Sunday' is highlighted. The main table shows 9 rows of sample data. The 'Properties' pane on the right shows the query name as 'qryColumnFromExamples'. The 'Applied Steps' pane shows the 'Column From Examples' step applied.

**Figure 53:** After typing “Sunday” as the first desired output, Power Query figures out the pattern.



7. The weekday name is correctly populated for all the remaining dates. Power Query figured out the pattern immediately. At the top of the screen, the resultant M language formula is shown.

At this point, if we were to click **OK**, the new column would be added to the Query.

## 7.2 Custom date format

Now we will apply a different transformation to the “Date” column: format as **dd mmm 'yy**.

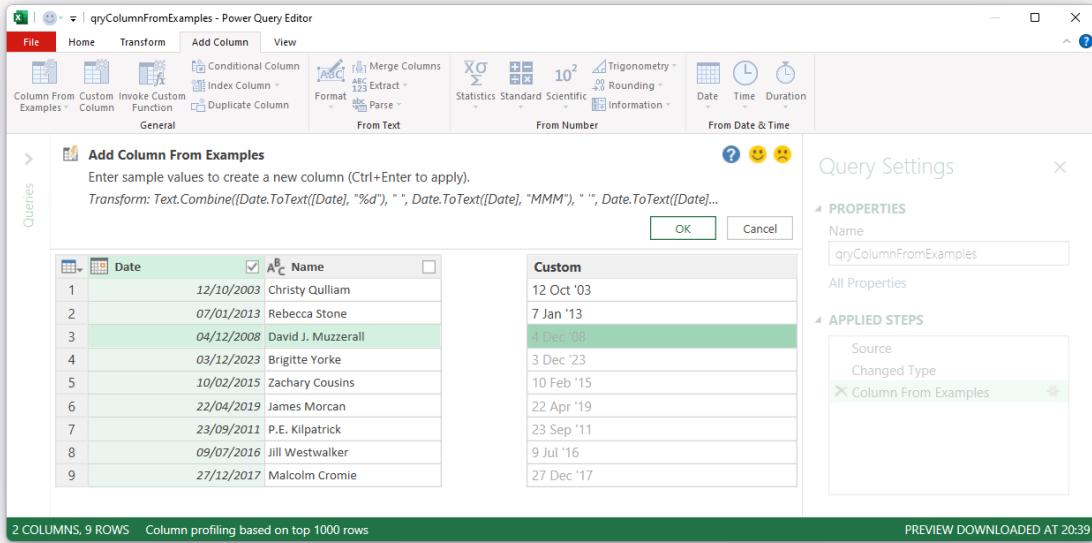
1. Provide the first example: “12 Oct '03”.

The screenshot shows the Power Query Editor interface. The 'Add Column From Examples' dialog is open, prompting for sample values to create a new column. A preview table shows the 'Date' and 'Name' columns with 9 rows of data. To the right, the 'Query Settings' pane displays the query name 'qryColumnFromExamples' and the applied steps, which include 'Column From Examples'.

Date	Name
12/10/2003	Christy Gulliam
07/01/2013	Rebecca Stone
04/12/2008	David J. Muzzell
03/12/2023	Brigitte Yorke
10/02/2015	Zachary Cousins
22/04/2019	James Morcan
23/09/2011	P.E. Kilpatrick
09/07/2016	Jill Westwalker
27/12/2017	Malcolm Cromie

**Figure 54:** After providing Column From Examples with one example.

2. In this instance, Power Query could not guess the pattern from only a single example. Try adding another example.



**Figure 55:** In this case, it took two examples for Power Query to infer the pattern.

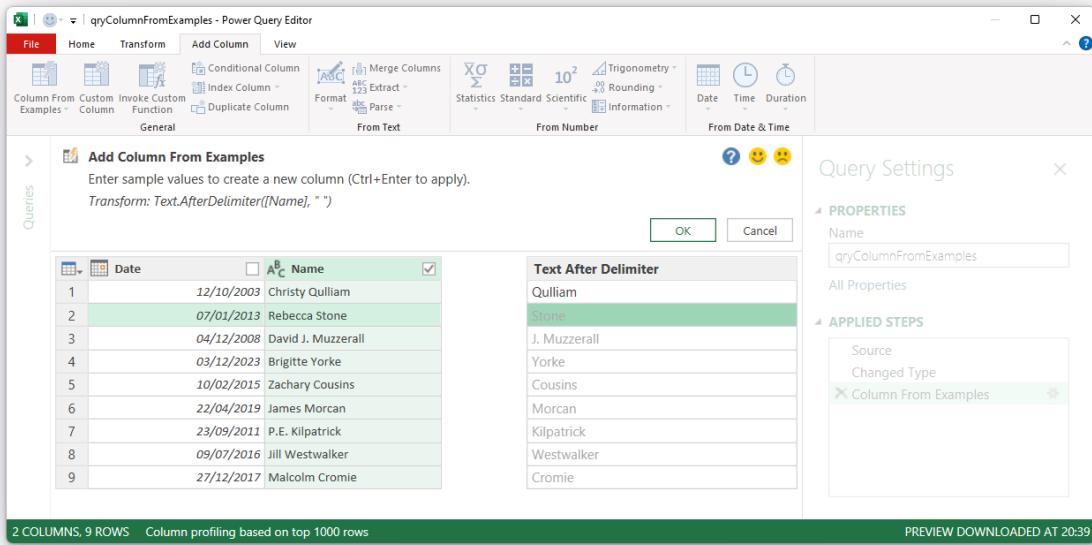
3. After the second example is provided, Power Query correctly infers the pattern.

## 7.3 Surnames

---

Column From Examples can perform text extraction easily. Let's extract the surnames.

1. In the **Column From Examples** interface, ensure that the “Name” column is checked and the “Date” column is unchecked.
2. Provide the first surname example: “Qulliam”.



**Figure 56:** Column From Examples makes it easy to perform simple string operations.

3. Power Query guesses the pattern immediately.

## 7.4 Multiple source columns

Finally, we will try to get Power Query to combine data from multiple columns.

1. Enter the first three desired outputs, as below.



The screenshot shows the Power Query Editor interface with the 'qryColumnFromExamples' query selected. In the 'Transform' ribbon, the 'Add Column' tab is active. A 'Column From Examples' dialog is open, titled 'Add Column From Examples'. It contains a preview table with two columns: 'Date' and 'Name'. The 'Name' column has a checkmark next to its header. Below the table is a 'Custom' section showing the results of the formula. The first four rows are populated with names from the source, while the remaining five rows are marked as 'null'. The 'Query Settings' pane on the right shows the query name 'qryColumnFromExamples' and the applied step 'Column From Examples'. The status bar at the bottom indicates '2 COLUMNS, 9 ROWS' and 'Column profiling based on top 1000 rows'.

**Figure 57:** When we combine data from multiple columns, Column From Examples might need a few more examples to guess the pattern.

2. We hoped that Power Query might propose “Yorke 2023”, “Cousins 2015”, etc., but instead we are getting **null**. Why?
3. Only columns marked with a checkmark are included in the formula for the new column. Check the “Date” column!

This screenshot is identical to Figure 57, except the 'Date' column in the preview table now has a yellow checkmark next to its header, indicating it is included in the formula. The 'Custom' section shows the resulting names with their respective years. The status bar at the bottom remains the same.

**Figure 58:** We must ensure that we enable all columns required in the formula.

4. As soon as we check the “Date” column, Power Query is able to produce the correct formula.



## Key Point

You may have noticed that the Applied Step associated with a Column From Examples command does **not** appear as “Column From Examples”. Instead, it appears as one of the other commands, such as “Added Custom Column” or “Added Conditional Column”. Power Query intelligently selects a command that produces the outputs we asked for.

## 7.5 Apply to student\_grades

Let’s return to the **student\_grades** Query. We will use **Column From Examples** to make a simple substitution in the “FamilySize” column.

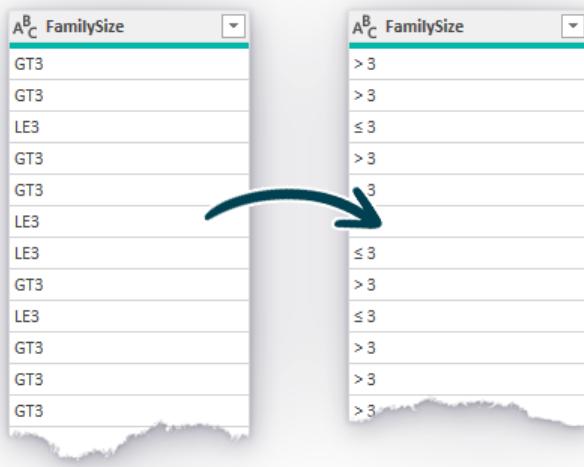
1. Inspect the “FamilySize” column. What do “GT3” and “LE3” stand for? You may remember this from earlier. If not, go back to the very first Applied Step and **read the metadata** for the “FamilySize” column.



### Helpful Hint

Remember: you can click a cell to view the full contents.

2. Use the **Column From Examples** tool to substitute “GT3” and “LE3” for something a little clearer. Use your best judgement.
3. When you are done, notice that an “Added Conditional Column” Step has been added. Power Query was smart enough to tell that our **Column From Examples** request could be simplified to a **Conditional Column** command.
4. Delete the old “FamilySize” column. Rename the new column “FamilySize”.



**Figure 59:** The new “FamilySize” column, created using Column From Examples.

## 7.6 Summary

---

- Column From Examples is an intelligent tool in Power Query that attempts to infer the command or formula that will produce the outputs we specify.
- Column From Examples can draw data from one or more columns.

# 8 Group and aggregate

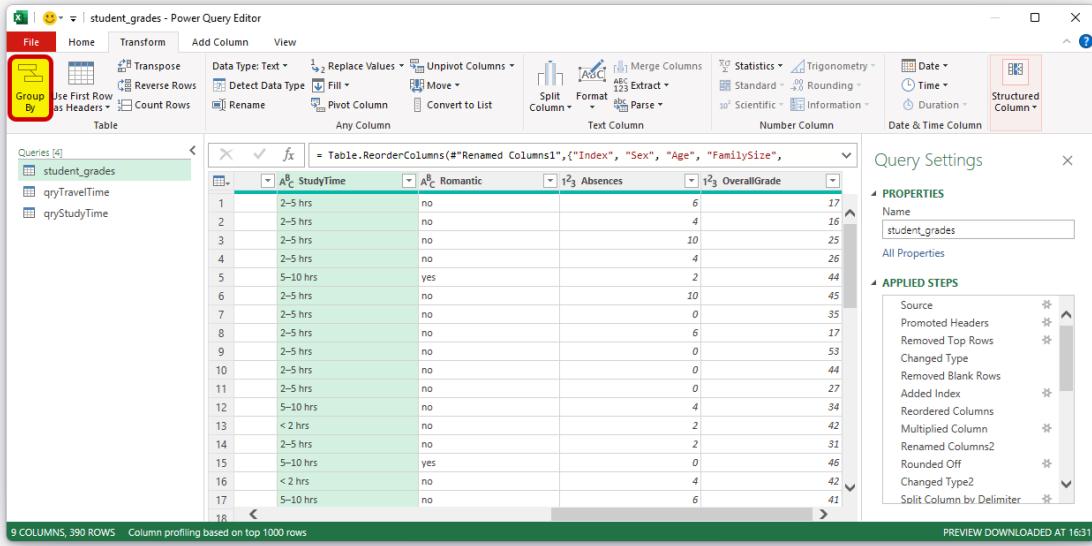
---

## 8.1 Is study time predictive of grade?

---

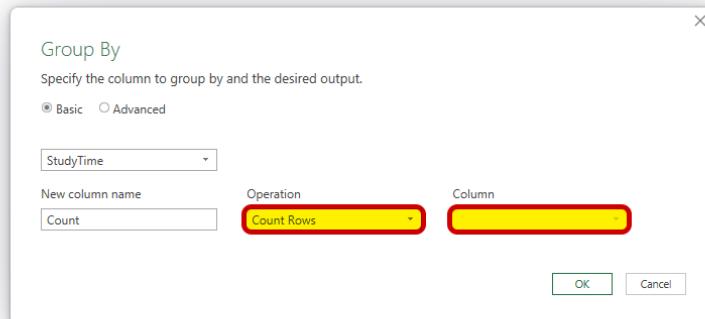
What is the mean of “OverallGrade” for each unique value of “StudyTime”? We can answer this question applying a **Group By** operation.

1. Select the “StudyTime” column. This is the column we want to group by.
2. Navigate to **Ribbon** > **Transform** tab > **Table** group > **Group By**.



**Figure 60:** Preparing to group by “StudyTime”.

3. The **Group By** window appears.



**Figure 61:** The **Group By** window.

4. Notice that the group-by column is “StudyTime”. If we wanted to, we could choose a different group-by column, or select “Advanced” to group by multiple columns. We will leave the group-by column as “StudyTime”.
5. Browse the options available under **Operation** and **Column**. Select options that will calculate the mean of “OverallGrade”. Specify a **New column name**. Click **OK**.

	A B C StudyTime	1.2 MeanGrade
1	2–5 hrs	31.35025381
2	5–10 hrs	34.95384615
3	< 2 hrs	30.69306931
4	> 10 hrs	35.18518519

**Figure 62:** The output of the Group By operation applied to the “StudyTime” column.

6. We can see that a longer weekly study time seems to correlate with a higher grade.



### Activity

Our table would be easier to read if it was correctly ordered by “StudyTime”. As an optional activity, figure out a way of implementing this sort.

Hint: the information we need may lie in another Query. How can we combine data from multiple Queries?

## 8.2 Is having a romantic partner predictive of grade?

To answer this question, we will again apply **Group By**. You will need to roll back your **student\_grades** Query so that it appears as in Figure 60.

Alternatively, you can edit the “Grouped Rows” Applied Step.

1. Find the mean grades of students who {have, do not have} a romantic partner. Your Query should appear as below.

	A B C Romantic	1.2 Count
1	no	32.87209302
2	yes	30.43181818

**Figure 63:** The output of the Group By operation applied to the “Romantic” column.



## Activity

In the Group By window, experiment with the “Advanced” option to group by two columns at once.

Can you find a group with an average grade greater than 40?

Can you find a group with an average grade less than 20?

## 9 Help and support

If you have any questions about this guide, or you wish to report a typo, please contact Andrew Alvin at [andrew.alvin@qa.com](mailto:andrew.alvin@qa.com).



## 10 Appendices

---

### Appendix 1: Notes on the Data Model

---

The **Data Model** is a powerful tool for linking related tables so that they can be queried as if they were a single, composite table.

We will not be using the Data Model in this apprenticeship. However, you may notice that Excel sometimes automatically adds your Queries to the Data Model. **This is harmless and will not interfere with any other operations.**

If you wish, you may remove each Query from the Data Model.

1. Navigate to the **Queries & Connections** panel > **right-click** a Query > **Load To...** > **uncheck** “Add this data to the Data Model”.
2. Repeat for all Queries.

If you do this, you will notice in the Queries & Connections panel that each Query now says “Connection only” below it instead of “<N> rows loaded”. This is a handy reminder that the Query has been loaded as a Connection, not a Table.