

Power Query Beginner Bootcamp

From messy data to clear insights with Excel & Power BI

Power Query ⚡ Basics



Course Overview

Power Query is Microsoft's data transformation and preparation engine built into Excel and Power BI. It lets you connect to diverse sources, shape messy data into clean, analysis-ready tables and load the results directly into your spreadsheet or model. The Beginner Bootcamp walks you through the full extract–transform–load (ETL) flow, showing how to launch the Power Query Editor, navigate its user interface and perform common tasks in Excel or Power BI Desktop.

The guide uses a repeatable, hands-on format: each module explains a concept, presents click-by-click steps, includes practice tasks with sample CSV and Excel files and ends with a short checklist. By the end, you'll know how to import data from workbooks, text/CSV files and folders, clean and standardize fields, reshape tables with pivot/unpivot and group-by, build append and merge queries, write simple M functions, and deliver a refreshable dataset ready for reporting.

Course Modules

- **Introduction & ETL Overview:** Learn what Power Query does and why it matters. Understand the Connect → Transform → Load workflow, the difference between Excel and Power BI destinations and how to open the editor.
- **User Interface & Applied Steps:** Identify the ribbon, queries pane, data preview and applied steps. See how each transformation is recorded as a step and why step order is crucial. Use query groups to organize staging and final outputs.

- **Importing Data:** Connect to workbooks, CSV/text files and entire folders. Navigate the source dialog, choose to transform or load, and handle credentials and privacy levels. Combining files from a folder automatically appends them for you.
- **Cleaning & Shaping:** Filter, sort, remove or rename columns, promote headers and assign proper data types. Replace blanks with nulls, standardize dates, numbers and Boolean fields, remove duplicates, and use Fill Down/Up.
- **Reshaping & Aggregating:** Unpivot wide tables into long format or pivot long tables into wide format. Group by one or more columns to create aggregations, create conditional columns (e.g., sales band) and custom columns such as `Number.Round([NetSales],0)`.
- **Calendar Table:** Build a reusable date dimension using M code. Starting with a start and end date, generate a list of dates and add year, quarter, month and day attributes. Adjust data types and sort order as needed.
- **Combining Data:** Append tables with the same schema or merge queries to bring columns from one table into another based on keys. Understand join types (Left, Inner, Right, Full) and expand only the columns you need.
- **Introduction to M:** Open the Advanced Editor to see the underlying M script created by the UI. Recognize the let/in structure and learn safe edits such as renaming variables, changing delimiters or replacing values. Use comments (//) to document your queries.
- **Real-World Exercise:** Bring everything together by building a monthly sales fact table from a folder of CSVs, enriching it with product details, adding calculated columns like GrossMargin and loading the finished dataset to Excel or Power BI. Refresh to pick up new files automatically.
- **Wrap-Up & Next Steps:** Review troubleshooting tips, best practices (clear naming, staging queries, loading only what you need) and where to learn more. An appendix provides cheat sheets for UI commands, common transformations, M patterns and error handling.

Calendar Table Example (M Code)

```
let
    // 1) Choose your date range
    StartDate = #date(2024, 1, 1),
    EndDate = #date(2026, 12, 31),

    // 2) Create a list of dates, then convert to a table
    DayCount = Duration.Days(EndDate - StartDate) + 1,
    DateList = List.Dates(StartDate, DayCount, #duration(1, 0, 0, 0)),
    DimDate0 = Table.FromList(DateList, Splitter.SplitByNothing(), {"Date"}),

    // 3) Add common calendar columns
    AddYear = Table.AddColumn(DimDate0, "Year", each Date.Year([Date]), Int64.Type),
    AddQuarter = Table.AddColumn(AddYear, "Quarter", each "Q" &
        Text.From(Date.QuarterOfYear([Date])), type text),
```

```
AddMonthNo      = Table.AddColumn(AddQuarter, "MonthNo", each Date.Month([Date]),
Int64.Type),
AddMonthName     = Table.AddColumn(AddMonthNo, "MonthName", each
Date.MonthName([Date]),
type text),
AddYearMonth     = Table.AddColumn(AddMonthName, "YearMonth", each Date.ToText([Date],
"yyyy-MM"),
type text),
AddWeek          = Table.AddColumn(AddYearMonth, "WeekOfYear", each
Date.WeekOfYear([Date]),
Int64.Type),
AddDay           = Table.AddColumn(AddWeek, "Day", each Date.Day([Date]), Int64.Type),
AddDayName       = Table.AddColumn(AddDay, "DayName", each Date.DayOfWeekName([Date]),
type text),
AddDOW           = Table.AddColumn(AddDayName, "DayOfWeek", each Date.DayOfWeek([Date],
Day.Monday)
+ 1,
Int64.Type),
AddIsWeekend     = Table.AddColumn(AddDOW, "IsWeekend", each Date.DayOfWeek([Date],
Day.Monday)
>= 5,
type logical),
AddStartMonth    = Table.AddColumn(AddIsWeekend, "StartOfMonth", each
Date.StartOfMonth([Date]),
type date),
AddEndMonth      = Table.AddColumn(AddStartMonth, "EndOfMonth", each
Date.EndOfMonth([Date]),
type date)
in
AddEndMonth
```

Who Should Attend

This bootcamp is designed for analysts, finance professionals, managers and anyone who spends hours cleaning and combining data in Excel. If you are new to Power Query or Power BI and want a structured introduction, this course is for you. It assumes only basic spreadsheet skills and teaches you how to replace manual copy-paste and VLOOKUP operations with repeatable, refreshable queries. Excel users looking to modernize their reporting workflows and Power BI beginners who need solid data foundations will benefit most.

What to Expect

Plan for approximately four to five hours of guided learning spread across multiple modules. You'll follow along using sample CSV and Excel files, performing each step yourself in Excel and Power BI. The curriculum progresses from simple imports and interface orientation to advanced shaping and combining techniques, culminating in a real-world exercise that produces a sales fact table and product dimension table ready for reporting. Throughout, you'll gain an appreciation for best practices: naming conventions, organizing staging vs final queries, careful step order and loading only the data you need. By the end, you'll be comfortable opening the Power Query Editor, executing common transformations, understanding the M script behind them and building refreshable datasets.