M Function and M Query in Power BI are part of the language used in Power Query, which is the data transformation engine for Power BI, Excel, and other Microsoft tools. Power Query uses the **M language** to create **queries** and transform data in various ways.

Here's a breakdown covering each concept with detailed explanations:

What is **M Function**?

M Functions are the building blocks of the M language (also called **Power Query Formula Language**). They allow you to perform data transformations, computations, and operations on your data within Power Query. M Functions are predefined methods in M that perform a specific action, such as adding, filtering, sorting, or merging data, as well as manipulating text, dates, and numbers.

Examples of common M functions:

- **Text functions**: For working with text (e.g., `Text.Upper`, `Text.Lower`, `Text.Combine`).
- **Date functions**: For handling dates and times (e.g., `Date.Year`, `Date.AddDays`).
- **List functions**: For working with lists, such as adding and removing elements (e.g., `List.Sum`, `List.Distinct`).
- **Table functions**: For manipulating tables (e.g., `Table.SelectRows`, `Table.AddColumn`).

What is **M Query**?

An **M Query** is a series of M code lines written in Power Query to transform and manipulate data. It's a sequence of steps that Power Query follows to import, clean, and shape your data. Each step in the Power Query Editor is recorded as an M Query command, which is executed in sequence to produce the final transformed data.

M Query Syntax is similar to other functional programming languages and is organized as a series of functions applied to your data. Power Query automatically generates M code for every action you perform in the editor, which you can view and edit if needed.

For example, this simple M Query renames columns, filters rows, and sorts data:

```
m
let
    Source = Excel.Workbook(File.Contents("C:\Data.xlsx"),
null, true),
    Sheet = Source{[Name="Sheet1"]}[Content],
    FilteredRows = Table.SelectRows(Sheet, each [Sales] >
1000),
    RenamedColumns = Table.RenameColumns(FilteredRows,
{{"Date", "Sales Date"}, {"Product", "Product Name"}}),
    SortedData = Table.Sort(RenamedColumns, {{"Sales Date",
Order.Ascending}})
in
    SortedData
```

What is `=Shared` in M Query?

`=Shared` is a special command in M Query that, when entered in Power Query's **Advanced Editor** or **Formula Bar**, lists all the built-in functions, operators, and constants available in Power Query. It's a way to explore the full library of available M functions.

To use `=Shared`:

1. Open Power Query Editor in Power BI.
2. Click on **Advanced Editor** or add a **New Query**.
3. In the formula bar, type `=Shared` and press **Enter**.

After running `=Shared`, you'll see a table that lists:

- **Function Names**: Names of all built-in M functions.
- **Descriptions**: Short descriptions of what each function does.

This is helpful for learning available functions and can be a valuable resource for finding specific functions to use in your queries.

Uses of M Functions

M Functions can be used in various ways to shape and transform data. Here are some practical examples of how to use M Functions in Power Query:

1. **Text Manipulation**

Example: Converting all text in a column to uppercase.

```
m
   Table.TransformColumns(Source, {{"Product", Text.Upper}})
```

This M Function converts all values in the **Product** column to uppercase.

2. **Date Calculations**

Example: Extracting the year from a date column.

```
m
   Table.AddColumn(Source, "Year", each
Date.Year([OrderDate]))
```

This adds a new column called **Year** to the table, with the year extracted from the **OrderDate** column.

3. **Conditional Columns**

Example: Creating a column that labels orders as "High" if Sales > 1000 and "Low" otherwise.

```
m
   Table.AddColumn(Source, "SalesCategory", each if [Sales] >
1000 then "High" else "Low")
```

This function creates a new column, **SalesCategory**, based on a conditional expression.

4. **Aggregations and Summarizations**

Example: Summing up the values in a column.

```
m
   List.Sum(Source[Sales])
```

This M Function calculates the sum of all values in the **Sales** column.

5. **Data Transformation with Lists and Records**

- **Lists** and **Records** are fundamental data structures in Power Query.
- **Lists** are ordered collections of values (e.g., `{1, 2, 3, 4}`).
- **Records** are collections of fields (similar to rows in a table) with name-value pairs (e.g., `[Name="John", Age=30]`).

  Example: Creating a list from a column and finding the maximum value.

  ```
  m
  List.Max(Source[Sales])
  ```
  This finds the maximum sales value in the **Sales** column.

6. **Creating Custom Functions**

You can define custom functions in M to simplify repetitive transformations.

Example: Defining a function to calculate profit.

```
m
   Profit = (price as number, cost as number) => price - cost
```

You can then use `Profit` as a reusable function in other parts of your M Query.

Basic Structure of an M Query

M Queries typically follow a `let...in` structure:

- `let` **block**: Defines a series of steps and assigns them to variables.
- `in` **statement**: Specifies the final output.

Example:

```
m
let
    Source = Table.FromRecords({[Name="John", Age=30],
[Name="Jane", Age=25]}),
    Filtered = Table.SelectRows(Source, each [Age] > 28),
    Renamed = Table.RenameColumns(Filtered, {{"Name", "Person
Name"}})
in
    Renamed
```

In this example:

1. The `Source` step creates a table from records.
2. The `Filtered` step filters rows where **Age** is greater than 28.
3. The `Renamed` step renames the **Name** column to **Person Name**.
4. The final result is `Renamed`.

Exploring More M Functions with Examples

Here are some more examples of frequently used M Functions:

`Table.AddColumn`

Adds a new column to an existing table.

```
m
Table.AddColumn(Source, "NewColumn", each [Column1] +
[Column2])
```

`Table.SelectRows`

Filters rows based on a condition.

```
m
Table.SelectRows(Source, each [Sales] > 500)
```

`Text.AfterDelimiter`

Extracts text after a specified delimiter.

```
m
Text.AfterDelimiter([Email], "@")  // Extracts domain from
email
```

`DateTime.LocalNow`

Gets the current date and time.

```
m
DateTime.LocalNow()
```

Summary Table of Key Points

Concept	Description	Example Usage
M Functions	Built-in functions in Power Query for data manipulation and transformation	`Text.Upper` , `Date.Year` , `Table.AddColumn`
M Query	Code that Power Query uses to perform a series of transformations on data	Defines steps like `Source` , `Filtered` , `Renamed`
=Shared	Displays a list of all available M functions in Power Query	Type `=Shared` in Advanced Editor
Uses of M Functions	Data manipulation, text processing, date calculations, custom calculations, conditional columns, aggregations	Summing values, creating conditional columns, formatting text
Basic Structure	`let...in` structure for organizing queries	Uses steps defined in `let` and returns result in `in`

By using **M Functions** and understanding **M Query** structure, you can unlock the full potential of Power Query to handle complex data transformations and gain more control over your data modeling process in Power BI.

In [ ]: