M language (also known as **Power Query M**) is a functional programming language used in **Power Query** within Power BI, Excel, and other Microsoft products. It is primarily designed for **data transformation, cleansing, and shaping** before loading data into Power BI.

◆ Key Features of M Language:

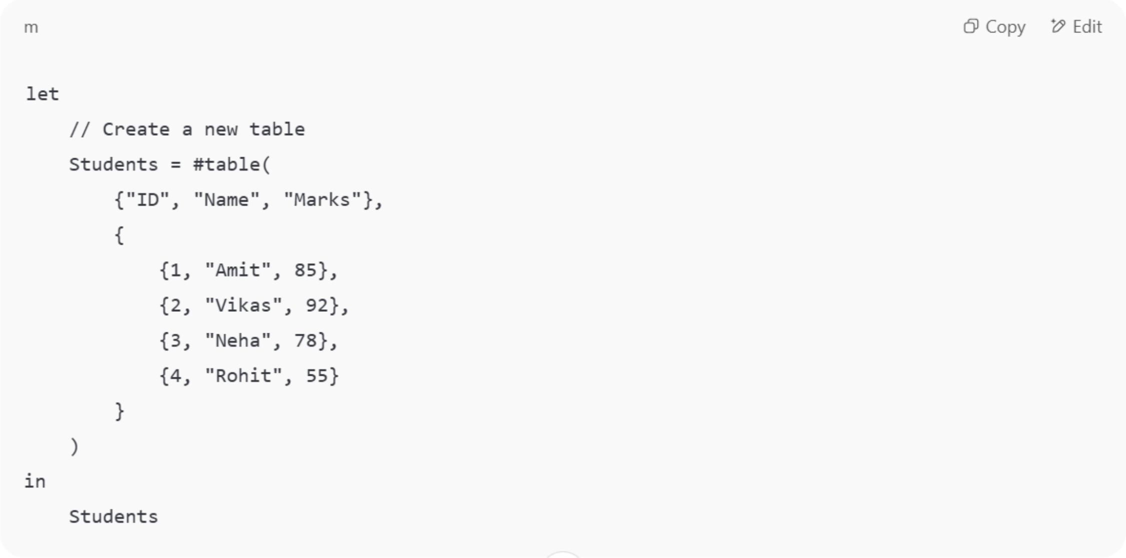
✅ **Case-Sensitive** – `Column` and `column` are treated differently.
✅ **Functional Language** – Uses functions for transformation (`Table.TransformColumns`, `Text.Combine`, etc.).
✅ **Step-Based Execution** – Each transformation is defined as a step in Power Query.
✅ **Immutable** – Once a variable is assigned, its value cannot change.

◆ Basic Syntax of M Language

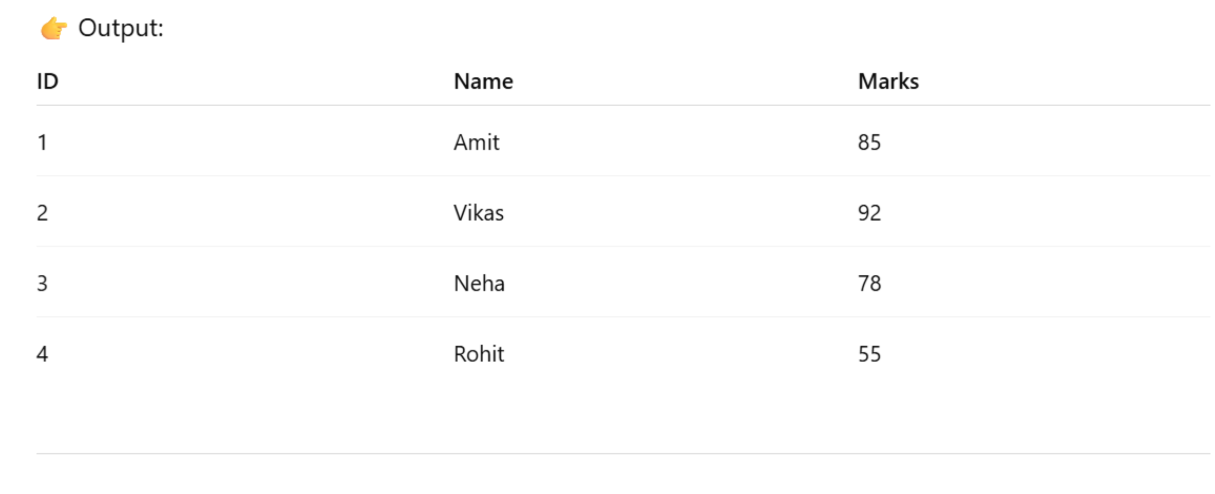
M queries start with the `let... in` construct.

Example – Student Marks Table

Step 1 – Create a New Table

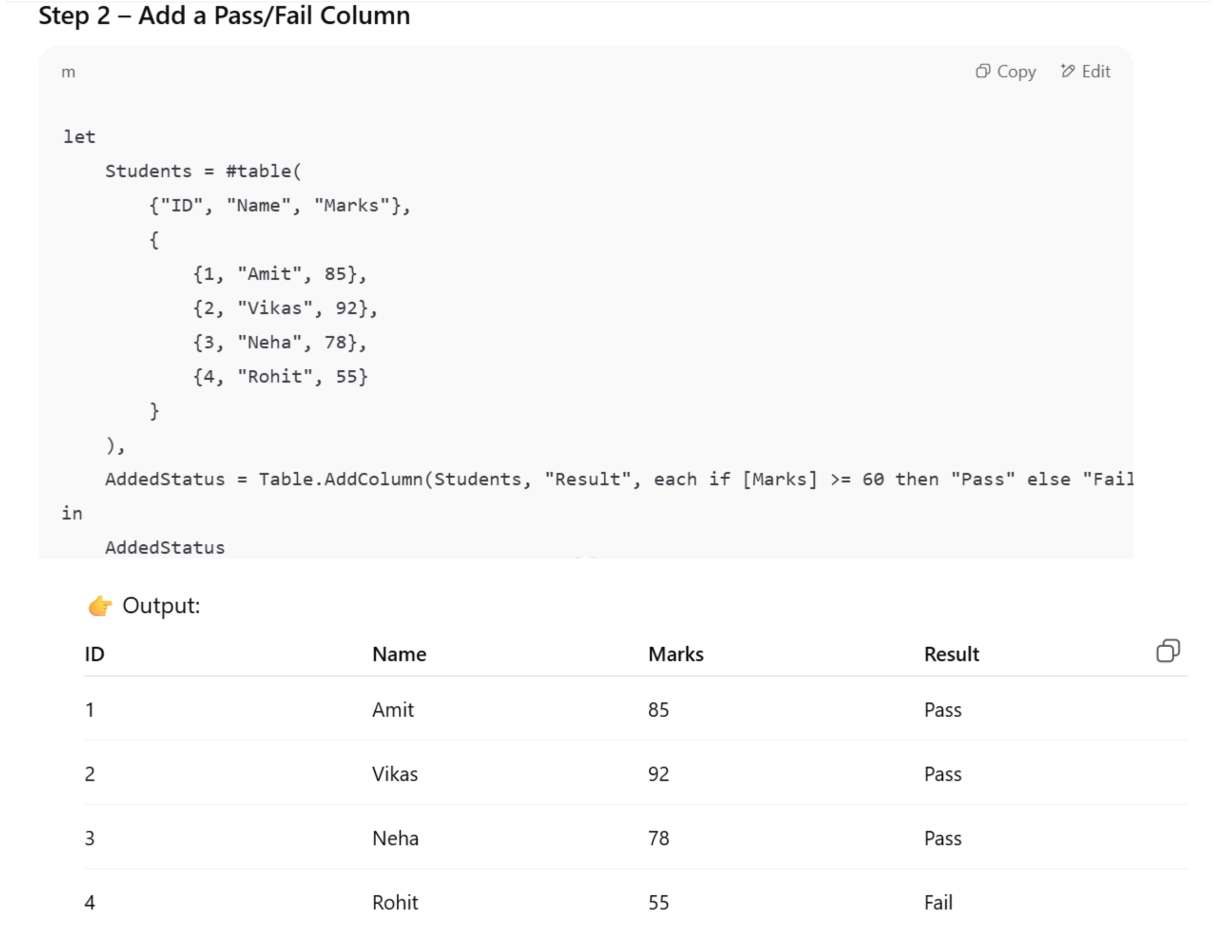


```m
let
    // Create a new table
    Students = #table(
        {"ID", "Name", "Marks"},
        {
            {1, "Amit", 85},
            {2, "Vikas", 92},
            {3, "Neha", 78},
            {4, "Rohit", 55}
        }
    )
in
    Students
```



👉 Output:

ID	Name	Marks
1	Amit	85
2	Vikas	92
3	Neha	78
4	Rohit	55



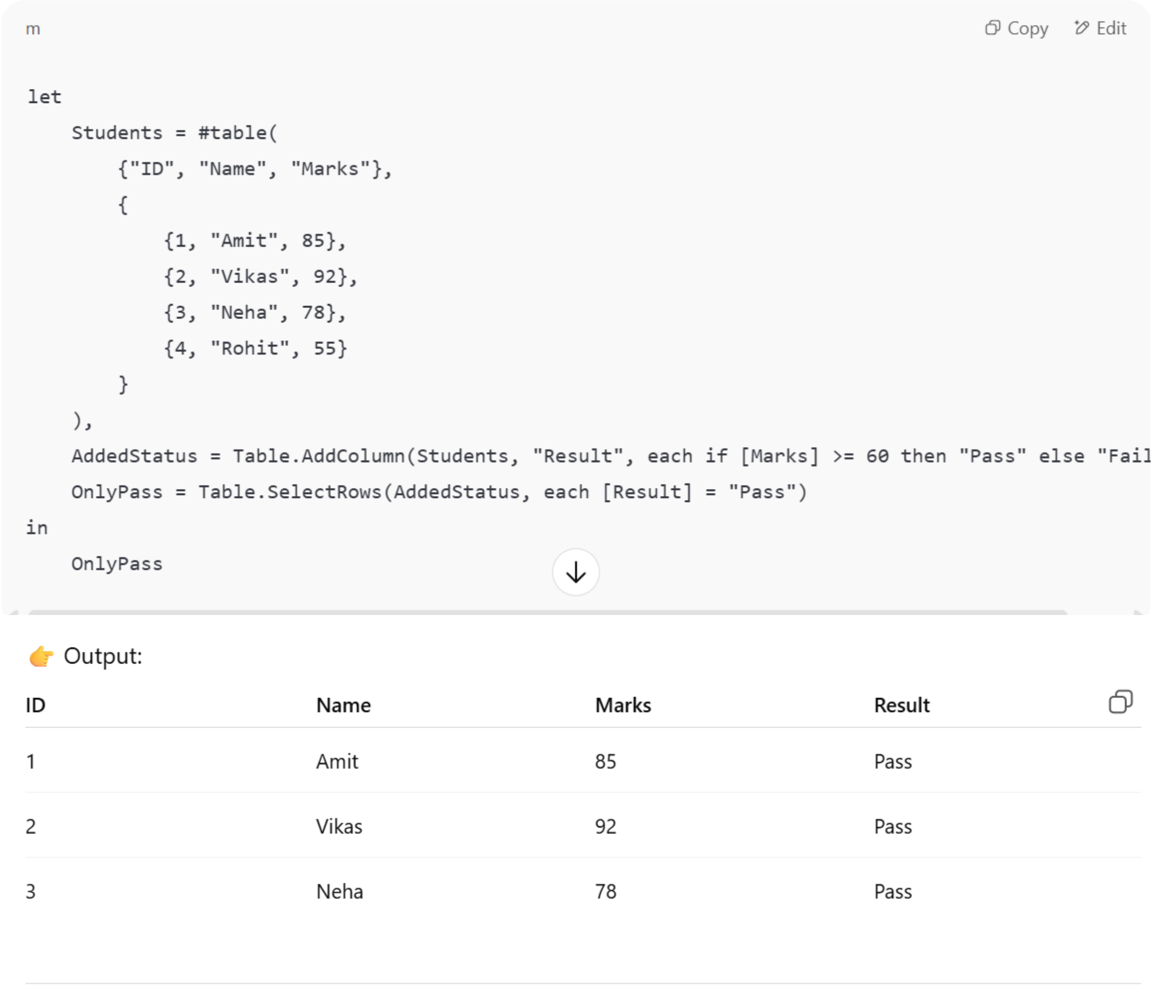
Step 2 – Add a Pass/Fail Column

```
let
    Students = #table(
        {"ID", "Name", "Marks"},
        {
            {1, "Amit", 85},
            {2, "Vikas", 92},
            {3, "Neha", 78},
            {4, "Rohit", 55}
        }
    ),
    AddedStatus = Table.AddColumn(Students, "Result", each if [Marks] >= 60 then "Pass" else "Fail
in
    AddedStatus
```

👉 Output:

ID	Name	Marks	Result
1	Amit	85	Pass
2	Vikas	92	Pass
3	Neha	78	Pass
4	Rohit	55	Fail

Step 3 – Filter Only Pass Students



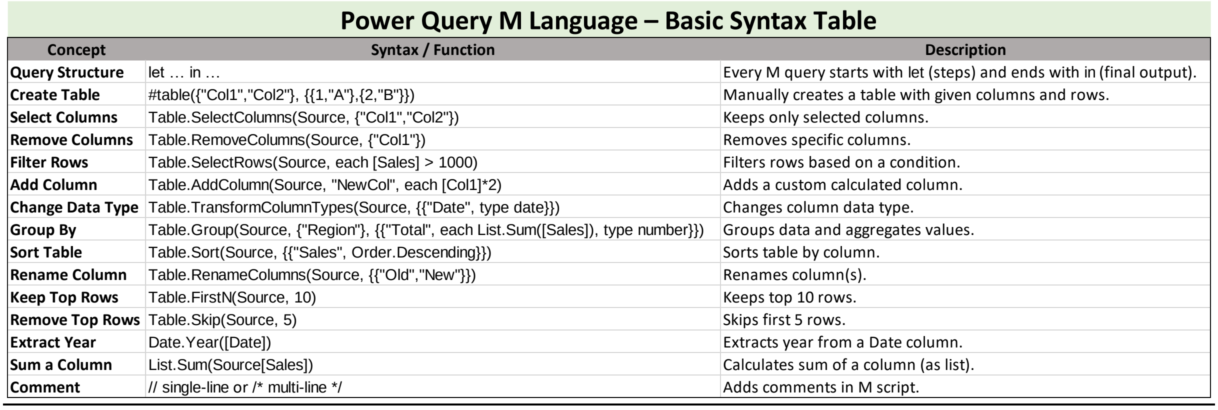
```
let
    Students = #table(
        {"ID", "Name", "Marks"},
        {
            {1, "Amit", 85},
            {2, "Vikas", 92},
            {3, "Neha", 78},
            {4, "Rohit", 55}
        }
    ),
    AddedStatus = Table.AddColumn(Students, "Result", each if [Marks] >= 60 then "Pass" else "Fail
    OnlyPass = Table.SelectRows(AddedStatus, each [Result] = "Pass")
in
    OnlyPass
```

👉 Output:

ID	Name	Marks	Result
1	Amit	85	Pass
2	Vikas	92	Pass
3	Neha	78	Pass

⚡ Key Takeaways:

- `#table` → creates a new table manually.
- `Table.AddColumn` → adds a new calculated column.
- `Table.SelectRows` → filters rows based on a condition.



Power Query M Language – Basic Syntax Table

Concept	Syntax / Function	Description
Query Structure	let … in …	Every M query starts with let (steps) and ends with in (final output).
Create Table	#table({"Col1","Col2"}, {{1,"A"},{2,"B"}})	Manually creates a table with given columns and rows.
Select Columns	Table.SelectColumns(Source, {"Col1","Col2"})	Keeps only selected columns.
Remove Columns	Table.RemoveColumns(Source, {"Col1"})	Removes specific columns.
Filter Rows	Table.SelectRows(Source, each [Sales] > 1000)	Filters rows based on a condition.
Add Column	Table.AddColumn(Source, "NewCol", each [Col1]*2)	Adds a custom calculated column.
Change Data Type	Table.TransformColumnTypes(Source, {{"Date", type date}})	Changes column data type.
Group By	Table.Group(Source, {"Region"}, {{"Total", each List.Sum([Sales]), type number}})	Groups data and aggregates values.
Sort Table	Table.Sort(Source, {{"Sales", Order.Descending}})	Sorts table by column.
Rename Column	Table.RenameColumns(Source, {{"Old","New"}})	Renames column(s).
Keep Top Rows	Table.FirstN(Source, 10)	Keeps top 10 rows.
Remove Top Rows	Table.Skip(Source, 5)	Skips first 5 rows.
Extract Year	Date.Year([Date])	Extracts year from a Date column.
Sum a Column	List.Sum(Source[Sales])	Calculates sum of a column (as list).
Comment	// single-line or /* multi-line */	Adds comments in M script.