[Power BI Data Transformation] [cheatsheet]

1. Data Import

- Import data from a CSV file: `Csv.Document(File.Contents("C:\data.csv"))`
- Import data from an Excel file: `Excel.Workbook(File.Contents("C:\data.xlsx"))`
- Import data from a JSON file: `Json.Document(File.Contents("C:\data.json"))`
- Import data from a SQL Server database: `Sql.Database("server", "database", [Query="SELECT * FROM table"])`
- Import data from a web API: `Web.Contents("https://api.example.com/data")`
- Import data from a folder: `Folder.Files("C:\data")`
- Import data from a SharePoint list: `SharePoint.Tables("https://contoso.sharepoint.com/sites/mysite", "ListName")`
- Import data from an OData feed: `OData.Feed("https://services.odata.org/V4/Northwind/Northwind.svc")`
- Import data from an ODBC source: `Odbc.Query("dsn=MyDSN", "SELECT * FROM table")`
- Import data from a text file: `Text.FromBinary(File.Contents("C:\data.txt"))`

2. Data Cleansing

- Remove duplicate rows: `Table.Distinct(table)`
- Remove blank rows: `Table.SelectRows(table, each not List.IsEmpty(Record.FieldValues(_)))`
- Replace null values with a specific value: `Table.ReplaceValue(table, null, 0, Replacer.ReplaceValue, {"ColumnName"})`
- Remove rows with errors: `Table.RemoveRowsWithErrors(table)`
- Fill down missing values: `Table.FillDown(table, {"ColumnName"})`
- Fill up missing values: `Table.FillUp(table, {"ColumnName"})`
- Trim whitespace from text: `Table.TransformColumns(table, {{"ColumnName", Text.Trim, type text}})`
- Clean text by removing non-printable characters: `Table.TransformColumns(table, {{"ColumnName", Text.Clean, type text}})`
- Capitalize text: `Table.TransformColumns(table, {{"ColumnName", Text.Proper, type text}})`

By: Waleed Mousa

- Lowercase text: Table.TransformColumns(table, {{"ColumnName", Text.Lower, type text}})
- Uppercase text: Table.TransformColumns(table, {{"ColumnName", Text.Upper, type text}})

3. Data Transformation

- Rename columns: Table.RenameColumns(table, {"OldColumnName", "NewColumnName"})
- Reorder columns: Table.ReorderColumns(table, {"Column1", "Column2", "Column3"})
- Remove columns: Table.RemoveColumns(table, {"ColumnName"})
- Filter rows based on a condition: Table.SelectRows(table, each [ColumnName] > 10)
- Sort rows: Table.Sort(table, {{"ColumnName", Order.Ascending}})
- Group rows and aggregate: Table.Group(table, {"GroupColumnName"}, {{"AggregateColumnName", each List.Sum([ColumnName]), type number}})
- Pivot data: Table.Pivot(table, List.Distinct(table[PivotColumnName]), "PivotColumnName", "ValueColumnName", List.Sum)
- Unpivot data: Table.UnpivotOtherColumns(table, {"ColumnName"}, "AttributeColumn", "ValueColumn")
- Transpose a table: Table.Transpose(table)
- Split a column by delimiter: Table.SplitColumn(table, "ColumnName", Splitter.SplitTextByDelimiter(","), {"Column1", "Column2"})
- Merge columns: Table.CombineColumns(table, {"Column1", "Column2"}, Combiner.CombineTextByDelimiter(" "), "NewColumnName")
- Extract text before a delimiter: Table.TransformColumns(table, {{"ColumnName", each Text.BeforeDelimiter(_, " "), type text}})
- Extract text after a delimiter: Table.TransformColumns(table, {{"ColumnName", each Text.AfterDelimiter(_, " "), type text}})
- Extract text between delimiters: Table.TransformColumns(table, {{"ColumnName", each Text.BetweenDelimiters(_, "{", "}"), type text}})
- Replace text: Table.TransformColumns(table, {{"ColumnName", each Text.Replace(_, "old", "new"), type text}})
- Add a custom column with a formula: Table.AddColumn(table, "NewColumnName", each [Column1] + [Column2], type number)
- Add an index column: Table.AddIndexColumn(table, "IndexColumn", 1, 1)
- Duplicate a column: Table.DuplicateColumn(table, "ColumnName", "NewColumnName")
- Merge queries: Table.NestedJoin(table1, {"JoinColumn"}, table2, {"JoinColumn"}, "NewColumnName", JoinKind.LeftOuter)

- Append queries: Table.Combine({table1, table2})

4. Aggregation and Grouping

- Count rows: Table.RowCount(Table)
- Sum a column: List.Sum(Table[ColumnName])
- Average a column: List.Average(Table[ColumnName])
- Find the minimum value in a column: List.Min(Table[ColumnName])
- Find the maximum value in a column: List.Max(Table[ColumnName])
- Group by a column and count rows: Table.Group(Table, {"ColumnName"}, {{"Count", each Table.RowCount(_), type number}})
- Group by a column and sum values: Table.Group(Table, {"ColumnName"}, {{"Sum", each List.Sum([ColumnToSum]), type number}})
- Group by a column and average values: Table.Group(Table, {"ColumnName"}, {{"Average", each List.Average([ColumnToAverage]), type number}})
- Group by a column and find the minimum value: Table.Group(Table, {"ColumnName"}, {{"Min", each List.Min([ColumnToMin]), type number}})
- Group by a column and find the maximum value: Table.Group(Table, {"ColumnName"}, {{"Max", each List.Max([ColumnToMax]), type number}})

5. Filtering and Sorting

- Filter rows based on a condition: Table.SelectRows(Table, each [ColumnName] > 10)
- Filter rows based on multiple conditions: Table.SelectRows(Table, each [Column1] > 10 and [Column2] = "Value")
- Filter rows based on a list of values: Table.SelectRows(Table, each List.Contains({"Value1", "Value2", "Value3"}, [ColumnName]))
- Filter rows based on a date range: Table.SelectRows(Table, each [Date] >= #date(2022, 1, 1) and [Date] <= #date(2022, 12, 31))
- Sort a table by a column in ascending order: Table.Sort(Table, {{"ColumnName", Order.Ascending}})
- Sort a table by a column in descending order: Table.Sort(Table, {{"ColumnName", Order.Descending}})
- Sort a table by multiple columns: Table.Sort(Table, {{"Column1", Order.Ascending}, {"Column2", Order.Descending}})

6. Joins and Merges

- Inner join two tables: Table.Join(Table1, "JoinColumn", Table2, "JoinColumn", JoinKind.Inner)
- Left outer join two tables: Table.Join(Table1, "JoinColumn", Table2, "JoinColumn", JoinKind.LeftOuter)

- Right outer join two tables: `Table.Join(Table1, "JoinColumn", Table2, "JoinColumn", JoinKind.RightOuter)`
- Full outer join two tables: `Table.Join(Table1, "JoinColumn", Table2, "JoinColumn", JoinKind.FullOuter)`
- Cross join two tables: `Table.CrossJoin(Table1, Table2)`
- Merge queries: `Table.NestedJoin(Table1, {"Key"}, Table2, {"ForeignKey"}, "NewColumn", JoinKind.LeftOuter)`
- Append queries: `Table.Combine({Table1, Table2})`

7. Date and Time Transformations

- Extract year from a date column: `Table.TransformColumns(table, {{"DateColumn", Date.Year, Int64.Type}})`
- Extract month from a date column: `Table.TransformColumns(table, {{"DateColumn", Date.Month, Int64.Type}})`
- Extract day from a date column: `Table.TransformColumns(table, {{"DateColumn", Date.Day, Int64.Type}})`
- Extract hour from a time column: `Table.TransformColumns(table, {{"TimeColumn", Time.Hour, Int64.Type}})`
- Extract minute from a time column: `Table.TransformColumns(table, {{"TimeColumn", Time.Minute, Int64.Type}})`
- Extract second from a time column: `Table.TransformColumns(table, {{"TimeColumn", Time.Second, Int64.Type}})`
- Extract day of week from a date column: `Table.TransformColumns(table, {{"DateColumn", Date.DayOfWeek, Int64.Type}})`
- Extract day of year from a date column: `Table.TransformColumns(table, {{"DateColumn", Date.DayOfYear, Int64.Type}})`
- Extract quarter from a date column: `Table.TransformColumns(table, {{"DateColumn", Date.QuarterOfYear, Int64.Type}})`
- Extract week of year from a date column: `Table.TransformColumns(table, {{"DateColumn", Date.WeekOfYear, Int64.Type}})`
- Add a specific number of days to a date column: `Table.TransformColumns(table, {{"DateColumn", each Date.AddDays(_, 7), type date}})`
- Add a specific number of months to a date column: `Table.TransformColumns(table, {{"DateColumn", each Date.AddMonths(_, 3), type date}})`
- Add a specific number of years to a date column: `Table.TransformColumns(table, {{"DateColumn", each Date.AddYears(_, 1), type date}})`

- Calculate the difference between two dates in days: Table.AddColumn(table, "DaysDiff", each Duration.Days([EndDate] - [StartDate]), type number)
- Calculate the difference between two times in hours: Table.AddColumn(table, "HoursDiff", each Duration.Hours([EndTime] - [StartTime]), type number)

8. Number Transformations

- Round numbers: Table.TransformColumns(table, {{"ColumnName", each Number.Round(_, 2), type number}})
- Truncate numbers: Table.TransformColumns(table, {{"ColumnName", each Number.Truncate(_), type number}})
- Ceiling numbers: Table.TransformColumns(table, {{"ColumnName", each Number.Ceiling(_), type number}})
- Floor numbers: Table.TransformColumns(table, {{"ColumnName", each Number.Floor(_), type number}})
- Absolute value of numbers: Table.TransformColumns(table, {{"ColumnName", each Number.Abs(_), type number}})
- Negate numbers: Table.TransformColumns(table, {{"ColumnName", each Number.Negate(_), type number}})
- Calculate the square root of numbers: Table.TransformColumns(table, {{"ColumnName", each Number.Sqrt(_), type number}})
- Calculate the logarithm of numbers: Table.TransformColumns(table, {{"ColumnName", each Number.Log(_), type number}})
- Calculate the exponential of numbers: Table.TransformColumns(table, {{"ColumnName", each Number.Exp(_), type number}})
- Calculate the modulo of numbers: Table.TransformColumns(table, {{"ColumnName", each Number.Mod(_, 5), type number}})
- Calculate the factorial of numbers: Table.TransformColumns(table, {{"ColumnName", each Number.Factorial(_), type number}})

9. Text Transformations

- Concatenate text columns: Table.AddColumn(table, "ConcatenatedColumn", each [Column1] & " " & [Column2], type text)
- Extract length of text: Table.TransformColumns(table, {{"ColumnName", each Text.Length(_), type number}})
- Extract first N characters from text: Table.TransformColumns(table, {{"ColumnName", each Text.Start(_, 5), type text}})

- Extract last N characters from text: Table.TransformColumns(table, {{"ColumnName", each Text.End(_, 5), type text}})
- Reverse text: Table.TransformColumns(table, {{"ColumnName", each Text.Reverse(_), type text}})
- Pad text with leading characters: Table.TransformColumns(table, {{"ColumnName", each Text.PadStart(_, 10, "0"), type text}})
- Pad text with trailing characters: Table.TransformColumns(table, {{"ColumnName", each Text.PadEnd(_, 10, "0"), type text}})
- Remove leading whitespace: Table.TransformColumns(table, {{"ColumnName", each Text.TrimStart(_), type text}})
- Remove trailing whitespace: Table.TransformColumns(table, {{"ColumnName", each Text.TrimEnd(_), type text}})
- Remove all whitespace: Table.TransformColumns(table, {{"ColumnName", each Text.Remove(_, " "), type text}})
- Replace text using a regular expression: Table.TransformColumns(table, {{"ColumnName", each Text.Replace(_, "[a-z]", "X"), type text}})

10. Conditional Transformations

- Add a conditional column: Table.AddColumn(table, "ConditionalColumn", each if [Column1] > 10 then "High" else "Low", type text)
- Filter rows based on a complex condition: Table.SelectRows(table, each if [Column1] > 10 and [Column2] = "A" then true else false)
- Apply a conditional formatting rule: Table.TransformColumns(table, {{"ColumnName", each if _ > 100 then "Green" else if _ > 50 then "Yellow" else "Red", type text}})
- Pivot data based on a condition: Table.Pivot(table, List.Distinct(table[PivotColumn]), "PivotColumn", "ValueColumn", each if _ = null then 0 else _)
- Group rows based on a condition: Table.Group(table, {"GroupColumn"}, {{"Count", each Table.RowCount(_), type number}, {"Sum", each if [ConditionColumn] = "A" then List.Sum([ValueColumn]) else 0, type number}})

11. Advanced Transformations

- Create a custom function: (x) => x * 2
- Apply a custom function to a column: Table.TransformColumns(table, {{"ColumnName", each MyCustomFunction(_), type number}})

- Apply a custom function to multiple columns: Table.TransformColumns(table, {{"Column1", each MyCustomFunction(_), type number}, {"Column2", each MyCustomFunction(_), type number}})
- Create a parameterized query: (parameter) => let Source = Csv.Document(File.Contents("C:\data.csv"), [Delimiter=","]), FilteredRows = Table.SelectRows(Source, each [ColumnName] = parameter) in FilteredRows
- Invoke a parameterized query: MyParameterizedQuery("ParameterValue")
- Create a function to merge multiple files: (folder) => let Source = Folder.Files(folder), CombinedData = Table.Combine(Source[Content]) in CombinedData
- Create a function to unpivot multiple columns: (table, columnsToUnpivot) => let UnpivotedTable = Table.UnpivotOtherColumns(table, columnsToUnpivot, "Attribute", "Value") in UnpivotedTable

12. Error Handling

- Replace errors with a specific value: Table.ReplaceErrorValues(table, {{"ColumnName", "DefaultValue"}})
- Remove rows with errors: Table.RemoveRowsWithErrors(table)
- Handle errors using a try-otherwise expression: Table.TransformColumns(table, {{"ColumnName", each try _ otherwise "Error", type text}})
- Handle missing or null values: Table.TransformColumns(table, {{"ColumnName", each if _ = null then 0 else _, type number}})

13. Data Profiling and Quality Assessment

- Count the number of rows: Table.RowCount(table)
- Count the number of distinct values in a column: Table.RowCount(Table.Distinct(table[ColumnName]))
- Find the minimum value in a column: List.Min(table[ColumnName])
- Find the maximum value in a column: List.Max(table[ColumnName])
- Calculate the average of a column: List.Average(table[ColumnName])
- Calculate the sum of a column: List.Sum(table[ColumnName])
- Calculate the standard deviation of a column: Table.StandardDeviation(table[ColumnName])
- Calculate the variance of a column: Table.Variance(table[ColumnName])
- Calculate the median of a column: Table.Median(table[ColumnName])
- Calculate the mode of a column: Table.Mode(table[ColumnName])
- Calculate the percentile of a column: Table.Percentile(table[ColumnName], 0.9)