



Power BI

250+ Formulas with Examples

Basic Aggregations

Learn More at myai.in

- MAX(): MAX(Profit) -- Latest/largest value
- COUNT(): COUNT(Order ID) -- Number of non-blank rows
- COUNTA(): COUNTA(Comments) -- Count including text/blank
- DISTINCTCOUNT(): DISTINCTCOUNT(Customer ID) -- Count unique values
- COUNTROWS(): COUNTROWS(Orders) -- Rows in a table expression
- COUNTX(): COUNTX(Products, Products[OnHand]) -- Iterator counting per row expression
- DIVIDE(): DIVIDE(Sales, Orders) -- Safe division handling zero
- PRODUCT(): PRODUCT(Quantity) -- Multiply values together
- PERCENTILE.EXC(): PERCENTILE.EXC(Sales,0.9) -- Exclusive percentile value
- MEDIAN(): MEDIAN(Profit) -- Median value within context
- STDEV.P(): STDEV.P(Sales) -- Population standard deviation
- VAR.S(): VAR.S(Profit) -- Sample variance

Date & Time

- TODAY(): TODAY() -- Current date per model timezone
- NOW(): NOW() -- Current date & time
- YEAR(): YEAR(Orders[Order Date]) -- Extract year component
- MONTH(): MONTH(Orders[Order Date]) -- Month number
- DAY(): DAY(Orders[Order Date]) -- Day of month
- HOUR(): HOUR(Orders[Order DateTime]) -- Hour part
- EOMONTH(): EOMONTH(Orders[Order Date],0) -- Last day of month offset
- DATE(): DATE(2025,4,23) -- Construct date from Y,M,D
- DATEVALUE(): DATEVALUE(Orders[DateText]) -- Convert text to date
- DATEDIFF(): DATEDIFF(Orders[Order Date], Orders[Ship Date], DAY) -- Difference between two dates
- DATEADD(): DATEADD(Dates[Date], -1, YEAR) -- Shift date column by interval
- SAMEPERIODLASTYEAR(): SAMEPERIODLASTYEAR(Dates[Date]) -- Parallel period previous year
- PARALLELPERIOD(): PARALLELPERIOD(Dates[Date], -3, MONTH) -- Parallel offset period
- TOTALYTD(): TOTALYTD(SUM(Sales), Dates[Date]) -- Year-to-date running total
- WEEKNUM(): WEEKNUM(Orders[Order Date]) -- ISO week number

Filter & Context

Learn More at myai.in

- CALCULATE(): CALCULATE(SUM(Sales), Year = 2025) -- Modify filter context for measure
- CALCULATETABLE(): CALCULATETABLE(Orders, Orders[Status]="Pending") -- Return table with new filters
- FILTER(): FILTER(Orders, Orders[Sales]>1000) -- Row-by-row filter returning table
- ALL(): ALL(Orders) -- Remove filters from specified table
- ALLEXCEPT(): ALLEXCEPT(Orders, Orders[Region]) -- Clear filters except listed columns
- ALLSELECTED(): ALLSELECTED(Orders) -- Respect report/page/visual selections
- REMOVEFILTERS(): REMOVEFILTERS(Orders[Segment]) -- Explicitly clear filters
- KEEPFILTERS(): KEEPFILTERS(Orders, Orders[Region]="East") -- Add filter without overwriting existing
- USERRELATIONSHIP(): CALCULATE(SUM(Sales), USERRELATIONSHIP(Date[Date],ShipDates[Date])) -- Activate inactive relationship
- CROSSFILTER(): CALCULATE(SUM(Sales), CROSSFILTER(Product[ID],Sales[PID],Both)) -- Override cross-filter direction
- VALUES(): VALUES(Orders[State]) -- Unique list of values in context
- DISTINCT(): DISTINCT(Orders[Category]) -- Table of unique values
- EARLIER(): VAR _t = EARLIER(Orders[Sales]) -- Reference row context from outer row
- RELATED(): RELATED(Product[Category]) -- Fetch related value from 1-side table
- RELATEDTABLE(): RELATEDTABLE(Sales) -- Return rows in related many-side table

Iterator & Table Functions

Learn More at imyai.in

- SUMX(): SUMX(Orders, Orders[Quantity] * Orders[UnitPrice]) -- Row-wise calculation then aggregation
- AVERAGEX(): AVERAGEX(Customers, Customers[Sales]) -- Iterator for average
- MINX(): MINX(Products, Products[Margin]) -- Iterator for min
- MAXX(): MAXX(Products, Products[Margin]) -- Iterator for max
- COUNTAX(): COUNTAX(Products, Products[Name]) -- Iterator counting non-blank strings
- RANKX(): RANKX(ALL(Customer), CALCULATE(SUM(Sales))) -- Rank measure within set
- TOPN(): TOPN(5, Customers, [Total Sales]) -- Return top N rows by expression
- BOTTOMN(): BOTTOMN(3, Products, [Profit]) -- Bottom N rows
- ADDCOLUMNS(): ADDCOLUMNS(Products, "Profit", Products[Sales]-Products[Cost]) -- Add calculated column to table
expression
- SELECTCOLUMNS(): SELECTCOLUMNS(Sales, "PID", Sales[ProductID], "Amount", Sales[Sales]) -- Return custom column
set
- UNION(): UNION(TableA, TableB) -- Combine tables vertically
- INTERSECT(): INTERSECT(CustomersUSA, CustomersEU) -- Common rows between tables
- EXCEPT(): EXCEPT(MasterList, Completed) -- Rows in first not in second
- CROSSJOIN(): CROSSJOIN(Products, Colors) -- Cartesian product of tables
- GENERATE(): GENERATE(Customers, Orders) -- Combine row and related table into single table

Logical & Testing

Learn More at myai.in

- IF(): IF(Profit>0, "Positive","Negative") -- Conditional branching
- SWITCH(): SWITCH(TRUE(), Score>=90,"A", Score>=80,"B","Other") -- Multiple condition branches
- AND(): AND(FlagA, FlagB) -- Logical AND
- OR(): OR(FlagA, FlagB) -- Logical OR
- NOT(): NOT(IsComplete) -- Logical NOT
- TRUE(): TRUE() -- Boolean constant
- FALSE(): FALSE() -- Boolean constant
- COALESCE(): COALESCE(Value, 0) -- Return first non-blank argument
- ISBLANK(): ISBLANK(Notes) -- Check for blank
- ISNUMBER(): ISNUMBER(Value) -- Check if numeric
- ISTEXT(): ISTEXT(Value) -- Check if text
- ERROR(): IFERROR(Expression, 0) -- Handle errors gracefully
- HASONEVALUE(): HASONEVALUE(Customer[State]) -- Test for single value in context
- ISFILTERED(): ISFILTERED(Orders[Region]) -- Test if column has filter
- ISINSCOPE(): ISINSCOPE(Date[Month]) -- Test if column in current hierarchy level

Math & Trigonometry

- ABS(): ABS(Profit) -- Absolute value
- CEILING(): CEILING(Discount, 1) -- Round up to multiple
- FLOOR(): FLOOR(Discount) -- Round down
- ROUND(): ROUND(Sales, 2) -- Round to decimals
- TRUNC(): TRUNC(Price, 0) -- Truncate without rounding
- INT(): INT(Revenue) -- Return integer part
- POWER(): POWER(Sales, 2) -- Exponentiation
- SQRT(): SQRT(Variance) -- Square root
- LN(): LN(Sales) -- Natural logarithm
- EXP(): EXP(1) -- Euler's constant e^x
- MOD(): MOD(Index, 2) -- Modulo operation
- QUOTIENT(): QUOTIENT(Quantity, 12) -- Integer division result
- RAND(): RAND() -- Random number 0-1
- RANDBETWEEN(): RANDBETWEEN(1,100) -- Random integer
- SIGN(): SIGN(Profit) -- Return sign of number

Misc Extras

Learn More at myai.in

- DATEDIFF (MONTH): DATEDIFF(Earliest[Date], Latest[Date], MONTH) -- Months between two dates
- QUARTER(): QUARTER(Orders[Order Date]) -- Quarter number of date
- HASONEFILTER(): HASONEFILTER(Date[Year]) -- Check if column filtered
- USERCULTURE(): USERCULTURE() -- Return locale of user
- ISOWeekNum(): WEEKNUM(Date[Date], 21) -- ISO week number alternative
- ISERROR(): ISERROR([Measure]) -- Check if measure returns error
- ISEVEN(): ISEVEN(Data[Index]) -- Test even number
- ISODD(): ISODD(Data[Index]) -- Test odd number
- NETWORKDAYS(): NETWORKDAYS(Start, End) -- Working days between dates (custom func)
- EARLIER(): EARLIER(Table[Column]) -- Access outer row context (again)

Power Query - Text & Number

- Text.Upper: Text.Upper([Name]) -- Convert text to uppercase in Power Query
- Text.Lower: Text.Lower([Name]) -- Lowercase
- Text.Trim: Text.Trim([Column]) -- Remove spaces
- Text.Length: Text.Length([Name]) -- Length of string
- Text.Replace: Text.Replace([Desc],"old","new") -- Replace substring
- Number.Round: Number.Round([Price],2) -- Round numeric to decimals
- Number.Abs: Number.Abs([Delta]) -- Absolute number
- Date.From: Date.From(DateTime.LocalNow()) -- Convert DateTime to Date
- DateTime.LocalNow: DateTime.LocalNow() -- Current timestamp in PQ
- Csv.Document: Csv.Document(File.Contents(path)) -- Parse CSV file

Power Query - Table Ops

Learn More at myai.in

- Table.SelectRows: Table.SelectRows(Source, each [Sales] > 1000) -- Filter rows in Power Query
- Table.AddColumn: Table.AddColumn(Source, "Profit", each [Sales]-[Cost]) -- Add calculated column
- Table.RemoveColumns: Table.RemoveColumns(Source,{"Index"}) -- Drop unwanted columns
- Table.RenameColumns: Table.RenameColumns(Source,{"old","new"}) -- Rename columns
- Table.Sort: Table.Sort(Source,{"Date",Order.Descending}) -- Sort rows
- Table.Group: Table.Group(Source,{"Region"},"Total", each List.Sum([Sales]), type number) --

Group and aggregate

- Table.ExpandTableColumn: Table.ExpandTableColumn(Nested,"Orders","Sales") -- Expand nested tables

- Table.Pivot: Table.Pivot(Source, List.Distinct(Source[Year]), "Year","Sales", List.Sum) -- Pivot rows to columns

- Table.Unpivot: Table.UnpivotOtherColumns(Source,{"ID"},"Attribute","Value") -- Unpivot columns

- Table.Combine: Table.Combine({Table1,Table2}) -- Append tables

- Table.Distinct: Table.Distinct(Source) -- Remove duplicate rows

- Table.FillDown: Table.FillDown(Source,{"Column1"}) -- Fill nulls with above value

- Table.ReplaceValue: Table.ReplaceValue(Source,null,0,Replacer.ReplaceValue,{"Qty"}) -- Replace values

- List.Generate: List.Generate(()=>>0, each _<10, each _+1) -- Generate list programmatically

- Excel.Workbook: Excel.Workbook(File.Contents(path)) -- Import Excel sheets

Power Query - Extras

- Table.RemoveRows: Table.RemoveRows(Source,0,1) -- Remove first N rows

- Table.FillUp: Table.FillUp(Source,{"Col"}) -- Fill nulls with below value

- Table.Transpose: Table.Transpose(Source) -- Swap rows/columns

- Text.EndsWith: Text.EndsWith([Col],"Ltd") -- Check suffix in PQ

- Text.StartsWith: Text.StartsWith([Col],"Inc") -- Check prefix

- Number.IntegerDivide: Number.IntegerDivide(15,4) -- Return quotient

- Number.Mod: Number.Mod(15,4) -- Return remainder

- List.Sum: List.Sum([Values]) -- Sum list items

- List.Sort: List.Sort([Values]) -- Sort list

- List.Contains: List.Contains([Values], "A") -- Test membership

Text & Info

Learn More at myai.in

- CONCATENATE(): CONCATENATE(FirstName, " ", LastName) -- Join strings
- CONCATENATEX(): CONCATENATEX(VALUES(Product[Category]), Product[Category], ", ") -- Concatenate across rows
- LEFT(): LEFT(Product[SKU], 3) -- Extract left characters
- RIGHT(): RIGHT(Product[SKU], 2) -- Extract right characters
- MID(): MID(Product[SKU], 4, 2) -- Substring
- LEN(): LEN(Product[Name]) -- String length
- LOWER(): LOWER(Customer[Email]) -- Lowercase
- UPPER(): UPPER(Customer[City]) -- Uppercase
- TRIM(): TRIM(Product[Name]) -- Remove leading/trailing spaces
- REPLACE(): REPLACE(Product[Name], 1, 5, "New") -- Replace substring by position
- SUBSTITUTE(): SUBSTITUTE(Comment, "bad", "good") -- Replace substring by match
- FIND(): FIND("@", Email, 1, 0) -- Position of substring
- SEARCH(): SEARCH("Pro", Product[Name], 1, -1) -- Case-insensitive find
- FORMAT(): FORMAT(Sales, "#,##0.00") -- Number/date formatting
- BLANK(): BLANK() -- Return blank value

Stat & Ranking

- PERCENTILEX.INC(): PERCENTILEX.INC(Orders, Orders[Sales], 0.9) -- Inclusive percentile across table
- STDEVX.P(): STDEVX.P(Sales, Sales[Amount]) -- Population std dev iterator
- VARX.P(): VARX.P(Sales, Sales[Amount]) -- Population variance iterator
- GEOMEAN(): GEOMEAN(Product[Growth]) -- Geometric mean
- RANKX(): RANKX(ALL(Customer), [Total Sales]) -- Dense rank within set
- TOPN(): TOPN(10, Products, [Total Profit]) -- Return top 10 products table
- PERCENTILE.INC(): PERCENTILE.INC(Sales, 0.5) -- Median via inclusive percentile
- QUARTILE.INC(): QUARTILE.INC(Sales, 1) -- Quartile value
- MODE.SNGL(): MODE.SNGL(Orders[Ship Mode]) -- Most frequent value
- CORR(): CORR(Sales[Amount], Profit[Amount]) -- Correlation coefficient

Additional DAX

Learn More at myai.in

- CURRENCY(): CURRENCY(Sales) -- Convert numeric to currency data type
- FORMAT(): FORMAT(Date, "MMM yyyy") -- Custom format string
- UNICHAR(): UNICHAR(128512) -- Unicode character
- UNICHAR(): UNICHAR(128525) -- Multiple expressions for icons
- PATHLENGTH(): PATHLENGTH(HierarchyPath) -- Count items in PATH
- DATATABLE(): DATATABLE("Col",STRING,{ {"A"},{"B"} }) -- Inline static table
- GENERATESERIES(): GENERATESERIES(1,10,1) -- Create number table
- ADDMISSINGITEMS(): ADDMISSINGITEMS(SUMMARIZE(...)) -- Ensure all combinations appear
- CONTAINSROW(): CONTAINSROW({"East","West"}, Region[Name]) -- Check column value in set
- MATCHBY(): MATCHBY(Customers[Year], Customers[Segment]) -- Define drive columns for calculation group
- TOCSV(): TOCSV(VALUES(Product[Category])) -- Concatenate list as CSV text
- ISO.CEILING(): ISO.CEILING(Sales, 5) -- Round up per ISO rule
- COMBINEVALUES(): COMBINEVALUES("-", ColumnA, ColumnB) -- Create composite key string
- SELECTEDVALUE(): SELECTEDVALUE(Date[Year]) -- Return single selected value or blank
- GROUPBY + CURRENTGROUP(): GROUPBY(Sales, Sales[Region], "Avg", AVERAGEX(CURRENTGROUP(), Sales[Sales])) -- Nested calculations within grouping
- ROWS(): ROWS(Products) -- Alias COUNTROWS
- DATETIME(): DATETIME(2025,4,23,10,30,0) -- Create date & time scalar
- STDEV.P(): STDEV.P(Sales) -- Population stdev scalar
- RANK.EQ(): RANK.EQ([Sales]) -- Excel style rank (2024 new)
- RANK.AVG(): RANK.AVG([Sales]) -- Average ranking for ties

Advanced Context & Hierarchy

Learn More at myai.in

- VAR: VAR _sales = SUM(Sales) -- Define reusable variable
- RETURN: VAR a=1 RETURN a+1 -- Return result of expression
- SUMMARIZE(): SUMMARIZE(Orders, Orders[Region], "Sales", SUM(Orders[Sales])) -- Group table and aggregate
- GROUPBY(): GROUPBY(Orders, Orders[Region], "Avg", AVERAGEX(CURRENTGROUP(),Orders[Sales])) --

Advanced group by with CURRENTGROUP

- ALLNOBLANKROW(): ALLNOBLANKROW(Customer) -- Remove blank row added for model integrity
- TREATAS(): TREATAS(VALUES(Region[Name]), Sales[Region]) -- Apply custom filter table onto column
- LOOKUPVALUE(): LOOKUPVALUE(Rates[Rate], Rates[Key], Sales[Key]) -- Search table for corresponding value
- PATH(): PATH(Parent[ID], Parent[ParentID]) -- Hierarchy path text
- PATHCONTAINS(): PATHCONTAINS(PathCol, SelectedID) -- Check if id in path
- PATHITEM(): PATHITEM(PathCol, 2, INTEGER) -- Return item from hierarchy path

User & Security

- USERPRINCIPALNAME(): USERPRINCIPALNAME() -- Current user's UPN - use in RLS
- USERNAME(): USERNAME() -- Legacy username function
- ISFILTERED(): ISFILTERED(Security[Dept]) -- Check filters for security logic
- ROLE SECURITY: FILTER(Sales, Sales[Region]=USERPRINCIPALNAME()) -- Row-level security formula
- HASONEFILTER(): HASONEFILTER(Dept[Name]) -- Detect multiple selections for security
- SELECTEDVALUE(): SELECTEDVALUE(Dept[Name]) -- Return selected slicer value safely
- GROUPBY RLS: KEEPFILTERS(Sales, Sales[Dept]=Employee[Dept]) -- Maintain filter within RLS
- ISINScope(): ISINScope(Region[Country]) -- Check hierarchy scope
- CONTAINS(): CONTAINS(Sales, Sales[User], USERPRINCIPALNAME()) -- Return TRUE if table contains row
- LOOKUPVALUE() RLS: LOOKUPVALUE(Auth[Allowed],Auth[User],USERPRINCIPALNAME()) -- Table lookup for permissions