**1. Statistical Functions**

These functions perform statistical calculations like percentile, median, and standard deviation.

* **MEDIAN**: Returns the median value in a column.
* Median Sales = MEDIAN(Sales[SalesAmount])
* **STDEV.P / STDEV.S**: Calculates population or sample standard deviation.
* StdDev Sales = STDEV.P(Sales[SalesAmount])
* **PERCENTILE.EXC / PERCENTILE.INC**: Returns a percentile value, exclusive or inclusive.
* 90th Percentile = PERCENTILE.EXC(Sales[SalesAmount], 0.9)

**2. Mathematical Functions**

Used for basic arithmetic operations and mathematical calculations.

* **ROUND**: Rounds a number to the specified number of digits.
* Rounded Sales = ROUND(Sales[SalesAmount], 2)
* **MOD**: Returns the remainder after division.
* Remainder = MOD(Sales[OrderID], 5)
* **POWER**: Raises a number to a power.
* Squared Value = POWER(Sales[Value], 2)

**3. Parent-Child Functions**

Useful for working with hierarchical data like organizational structures.

* **PATH**: Returns a delimited string with the identifiers of all parents to the current item.
* Org Path = PATH(Employees[EmployeeID], Employees[ManagerID])
* **PATHLENGTH**: Returns the depth of the hierarchy.
* Org Depth = PATHLENGTH(Org Path)
* **LOOKUPVALUE**: Returns the value in a column for a row that meets one or more criteria.
* Manager Name = LOOKUPVALUE(Employees[Name], Employees[EmployeeID], Employees[ManagerID])

**4. Date and Time Functions**

Expand on time intelligence by enabling more granular date manipulations.

* **NOW**: Returns the current date and time.
* Current DateTime = NOW()
* **YEAR / MONTH / DAY**: Extracts the year, month, or day from a date.
* Order Year = YEAR(Sales[OrderDate])
* **DATEDIFF**: Calculates the difference between two dates.
* Days Since Order = DATEDIFF(Sales[OrderDate], TODAY(), DAY)

**5. Table Functions**

These work with tables and are often used in combination with filter functions.

* **ADDCOLUMNS**: Adds a calculated column to a table.
* Sales with Tax = ADDCOLUMNS(Sales, "Tax", Sales[SalesAmount] \* 0.15)
* **SUMMARIZE**: Creates a summary table for specified columns and aggregations.
* Summary Table = SUMMARIZE(Sales, Sales[Region], "Total Sales", SUM(Sales[SalesAmount]))
* **CROSSJOIN**: Returns the Cartesian product of two tables.
* Combined Table = CROSSJOIN(Products, Regions)

**6. Information Functions**

Used to test the state or type of data.

* **ISBLANK**: Checks if a value is blank.
* Blank Check = IF(ISBLANK(Sales[Discount]), "No Discount", "Has Discount")
* **ISNUMBER**: Checks if a value is a number.
* Is Number = ISNUMBER(Sales[OrderID])
* **ERROR**: Forces an error for testing or handling in advanced logic.
* Force Error = ERROR("This is a test error")

**7. Value Functions**

Operate on or return values.

* **BLANK**: Returns a blank value.
* Default Value = IF(ISBLANK(Sales[ReturnDate]), BLANK(), Sales[ReturnDate])
* **VALUE**: Converts text to a numeric value.
* Numeric Conversion = VALUE(Sales[TextNumber])
* **FORMAT**: Converts a value to text in a specific format.
* Formatted Date = FORMAT(Sales[OrderDate], "MM/DD/YYYY")

**8. Ranking and Sorting Functions**

Facilitate ranking and sorting within datasets.

* **TOPN**: Returns the top N rows of a table based on an expression.
* Top 5 Products = TOPN(5, Products, Products[Sales], DESC)
* **RANK.EQ**: Ranks items with ties getting the same rank.
* Rank by Sales = RANK.EQ(Sales[SalesAmount], Sales, Sales[SalesAmount], DESC)

**9. Logical and Conditional Functions**

Support advanced decision-making logic.

* **AND / OR**: Logical operators for combining conditions.
* High Sales and Recent = IF(AND(Sales[SalesAmount] > 1000, Sales[Year] = 2023), "Yes", "No")
* **NOT**: Negates a condition.
* Not High Sales = IF(NOT(Sales[SalesAmount] > 1000), "Low", "High")

**10. Iterative and Row Context Functions**

Useful when performing calculations row-by-row.

* **GENERATE**: Combines tables with row-wise logic.
* Combined = GENERATE(Products, CALCULATETABLE(Sales, Products[ProductID] = Sales[ProductID]))
* **X Functions (e.g., SUMX, AVERAGEX)**: Work row-by-row with expressions.
* Total Weighted Sales = SUMX(Sales, Sales[Quantity] \* Sales[Price])