
Debabrata Palit
/debabrata-palit03

DAX Functions

Data Analysis Expressions

TRANSFORM YOUR DATA INTO DECISIONS!


Debabrata Palit
/debabrata-palit03

What is DAX?

DAX (Data Analysis Expressions) is a formula language used in **Microsoft** tools like **Power BI**, **Excel Power Pivot**, and **SQL Server Analysis Services** (SSAS Tabular models). It is designed to help users create custom calculations on data models — such as creating measures, calculated columns, and calculated tables.

What Can You Do With DAX?

- Perform aggregations like SUM, AVERAGE, COUNT
- Create calculated measures (like profit margin, YoY growth)
- Build complex business logic (e.g., sales only for top customers)
- Filter and manipulate data using context-aware formulas

Why to Learn DAX?

- Essential for **data modeling** in Power BI
- Enables **advanced analytics** without external tools
- Helps you build **dynamic, interactive dashboards**

1. Aggregation Functions

DAX Formula Name	Syntax	Definition
AVERAGE	AVERAGE(<column>)	Returns the average (arithmetic mean) of the values in a column.
AVERAGEX	AVERAGEX(<table>, <expression>)	Returns the average of an expression evaluated over a table.
COUNT	COUNT(<column>)	Counts the number of values in a column (ignoring blank values).
COUNTAX	COUNTAX(<table>, <expression>)	Counts the number of rows that contain a value in an expression evaluated over a table.
COUNTBLANK	COUNTBLANK(<column>)	Counts the number of blank values in a column.
COUNTROWS	COUNTROWS(<table>)	Returns the number of rows in a table or table expression.
DISTINCTCOUNT	DISTINCTCOUNT(<column>)	Counts the number of distinct (unique) values in a column.
MAX	MAX(<column>)	Returns the largest value in a column.
MAXX	MAXX(<table>, <expression>)	Returns the largest value from an expression evaluated over a table.
MIN	MIN(<column>)	Returns the smallest value in a column.

MINX	MINX(<table>, <expression>)	Returns the smallest value from an expression evaluated over a table.
SUM	SUM(<column>)	Returns the sum of values in a column.
SUMAX	SUMAX(<table>, <expression>)	Returns the sum of an expression evaluated over a table.
SUMX	SUMX(<table>, <expression>)	Returns the sum of an expression evaluated over a table.
PRODUCT	PRODUCT(<column>)	Returns the product of all values in a column.
PRODUCTX	PRODUCTX(<table>, <expression>)	Returns the product of an expression evaluated over a table.
MEDIAN	MEDIAN(<column>)	Returns the median value of a column.
STDEV	STDEV(<column>)	Returns the sample standard deviation of a column.
STDEVP	STDEVP(<column>)	Returns the population standard deviation of a column.
VAR	VAR(<column>)	Returns the sample variance of a column.
VARP	VARP(<column>)	Returns the population variance of a column.

2. Date and Time Functions

DAX Formula Name	Syntax	Definition
DATE	DATE(<year>, <month>, <day>)	Returns the date value for the specified year, month, and day.

DATEDIFF	DATEDIFF(<start_date>, <end_date>, <interval>)	Returns the difference between two dates, in the specified interval (e.g., days, months, years).
DATEADD	DATEADD(<dates>, <number>, <interval>)	Returns a table that shifts the dates in a column by a specified number of intervals.
DAYS	DAYS(<end_date>, <start_date>)	Returns the number of days between two dates.
DAY	DAY(<date>)	Returns the day of the month from a date.
ENDOFMONTH	ENDOFMONTH(<dates>, <shift>)	Returns the last date of the month, with an optional offset.
ENDOFYEAR	ENDOFYEAR(<dates>, <shift>)	Returns the last date of the year, with an optional offset.
HOUR	HOUR(<time>)	Returns the hour of a given time.
MONTH	MONTH(<date>)	Returns the month of the year from a date.
MONTHNAME	MONTHNAME(<date>)	Returns the name of the month for a given date.
NOW	NOW()	Returns the current date and time.
QUARTER	QUARTER(<date>)	Returns the quarter of the year from a date.
STARTOFMONTH	STARTOFMONTH(<dates>)	Returns the first date of the month.
STARTOFYEAR	STARTOFYEAR(<dates>)	Returns the first date of the year.
TIME	TIME(<hour>, <minute>, <second>)	Returns a time value for the specified hour, minute, and second.

TIMEVALUE	TIMEVALUE(<time>)	Returns the time portion of a date-time value as a time value.
TODAY	TODAY()	Returns the current date.
YEAR	YEAR(<date>)	Returns the year from a date.
YEARFRAC	YEARFRAC(<start_date>, <end_date>)	Returns the year fraction between two dates.
SAMEPERIODLASTYEAR	SAMEPERIODLASTYEAR(<dates>)	Returns a table of the same period last year, based on the provided dates.
TOTALYTD	TOTALYTD(<expression>, <dates>, <filter>, <year_end_date>)	Calculates the year-to-date (YTD) value for an expression over a date range.
DATESBETWEEN	DATESBETWEEN(<dates>, <start_date>, <end_date>)	Returns a table of dates between the specified start and end date.
DATESINPERIOD	DATESINPERIOD(<dates>, <start_date>, <number_of_periods>, <interval>)	Returns a table of dates within the specified period.
DATESMTD	DATESMTD(<dates>)	Returns a table of dates for the month-to-date (MTD).
DATESQTD	DATESQTD(<dates>)	Returns a table of dates for the quarter-to-date (QTD).
DATESYTD	DATESYTD(<dates>)	Returns a table of dates for the year-to-date (YTD).
PREVIOUSMONTH	PREVIOUSMONTH(<dates>)	Returns a table of dates for the previous month.

NEXTMONTH	NEXTMONTH(<dates>)	Returns a table of dates for the next month.
PARALLELPERIOD	PARALLELPERIOD(<dates>, <number_of_intervals>, <interval>)	Returns a table of dates shifted by a specified number of intervals.
FIRSTDATE	FIRSTDATE(<dates>)	Returns the first date in a column or table.
LASTDATE	LASTDATE(<dates>)	Returns the last date in a column or table.
LASTNONBLANK	LASTNONBLANK(<dates>, <expression>)	Returns the last date in a column or table that contains a non-blank value for the expression.

3. Filter Functions

DAX Formula Name	Syntax	Definition
ALL	ALL(<table_or_column>)	Removes all filters from a table or column.
ALLSELECTED	ALLSELECTED(<table_or_column>)	Returns all values in a table or column, considering any filters in the report context.
ALLEXCEPT	ALLEXCEPT(<table>, <column1>, <column2>, ...)	Removes all filters in a table except the ones on specified columns.
REMOVEFILTERS	REMOVEFILTERS(<table_or_column>)	Removes filters from a table or column.
SELECTEDVALUE	SELECTEDVALUE(<column>, <default>)	Returns the value when a single value is selected in a column, otherwise returns the default value.

VALUES	VALUES(<column>)	Returns a one-column table that contains the unique values in a column.
FILTER	FILTER(<table>, <expression>)	Returns a table that contains only rows for which the expression evaluates to true.
CALCULATE	CALCULATE(<expression>, <filter1>, <filter2>, ...)	Evaluates an expression in a modified filter context.
CALCULATETABLE	CALCULATETABLE(<table>, <filter1>, <filter2>, ...)	Returns a table modified by the specified filters.
TREATAS	TREATAS(<table>, <column1>, <column2>, ...)	Treats one table as if it were another, with different column names and relationships.
USERELATIONSHIP	USERELATIONSHIP(<column1>, <column2>)	Specifies an alternative relationship to be used in the calculation.
CROSSFILTER	CROSSFILTER(<column1>, <column2>, <relationship>)	Defines a cross-filtering direction for a relationship.
KEEPFILTERS	KEEPFILTERS(<filter>)	Applies additional filters to the current context in a calculation.
ISFILTERED	ISFILTERED(<column>)	Returns TRUE if the column is being filtered, otherwise returns FALSE.
ISBLANK	ISBLANK(<value>)	Returns TRUE if a value is blank, otherwise returns FALSE.

4. Information Functions

DAX Formula Name	Syntax	Definition
ISBLANK	ISBLANK(<value>)	Returns TRUE if the value is blank, otherwise returns FALSE.
ISERROR	ISERROR(<expression>)	Returns TRUE if the expression results in an error, otherwise returns FALSE.
ISNONTEXT	ISNONTEXT(<value>)	Returns TRUE if the value is not text, otherwise returns FALSE.
ISNUMBER	ISNUMBER(<value>)	Returns TRUE if the value is a number, otherwise returns FALSE.
ISTEXT	ISTEXT(<value>)	Returns TRUE if the value is text, otherwise returns FALSE.
ISLOGICAL	ISLOGICAL(<value>)	Returns TRUE if the value is a logical (TRUE/FALSE), otherwise returns FALSE.

5. Logical Functions

DAX Formula Name	Syntax	Definition
AND	AND(<expression1>, <expression2>)	Returns TRUE if both expressions evaluate to TRUE, otherwise returns FALSE.
IF	IF(<logical_test>, <value_if_true>, <value_if_false>)	Returns one value if the condition is TRUE and another if the condition is FALSE.
IFERROR	IFERROR(<expression>, <value_if_error>)	Returns the value of an expression if it does not result in an error, otherwise returns a specified value.
NOT	NOT(<logical_expression>)	Reverses the result of a logical expression, returning TRUE if the expression is FALSE, and vice versa.
OR	OR(<expression1>, <expression2>)	Returns TRUE if any of the expressions evaluate to TRUE.
SWITCH	SWITCH(<expression>, <value1>, <result1>, ...)	Evaluates an expression against a list of values and returns the result corresponding to the first matching value.
TRUE	TRUE()	Returns the logical value TRUE.
FALSE	FALSE()	Returns the logical value FALSE.

ISINSCOPE	ISINSCOPE(<column>)	Returns TRUE if a column is in the current scope, otherwise returns FALSE.
IFBLANK	IFBLANK(<value>, <alternative_value>)	Returns a value if the expression is not blank; otherwise, returns an alternative value.
COALESCE	COALESCE(<expression 1>, <expression2>, ...)	Returns the first non-blank value among the expressions.
IFERROR	IFERROR(<expression>, <value_if_error>)	Returns a specified value if an expression evaluates to an error.

6. Statistical Functions

DAX Formula Name	Syntax	Definition
MEDIAN	MEDIAN(<column>)	Returns the median value of a column.
PERCENTILE.EXC	PERCENTILE.EXC(<column>, <percentile>)	Returns the value at the given percentile of a column, excluding the 0th and 100th percentiles.
PERCENTILE.INC	PERCENTILE.INC(<column>, <percentile>)	Returns the value at the given percentile of a column, including the 0th and 100th percentiles.

DAX Formula Name	Syntax	Definition
STDEV	STDEV(<column>)	Returns the sample standard deviation of a column.
STDEVP	STDEVP(<column>)	Returns the population standard deviation of a column.
VAR	VAR(<column>)	Returns the sample variance of a column.
VARP	VARP(<column>)	Returns the population variance of a column.
RANKX	RANKX(<table>, <expression>, <value>, <order>, <ties>)	Returns the rank of an expression evaluated over a table, with options for sorting and handling ties.
RUNNINGTOTAL	RUNNINGTOTAL(<expression>, <dates>, <order>)	Computes the running total of an expression over time, based on the provided date and sort order.
COVARIANCE.P	COVARIANCE.P(<column1>, <column2>)	Returns the population covariance between two columns.
COVARIANCE.S	COVARIANCE.S(<column1>, <column2>)	Returns the sample covariance between two columns.

7. Text Functions

DAX Formula Name	Syntax	Definition
CONCATENATE	CONCATENATE(<text1>, <text2>)	Joins two text values into one text string.
CONCATENATEX	CONCATENATEX(<table>, <expression>, <delimiter>, <order>)	Concatenates the result of an expression evaluated over a table with a specified delimiter, and an optional sort order.
LEFT	LEFT(<text>, <num_chars>)	Returns the leftmost characters from a text string, based on a specified number of characters.
RIGHT	RIGHT(<text>, <num_chars>)	Returns the rightmost characters from a text string, based on a specified number of characters.
MID	MID(<text>, <start_position>, <num_chars>)	Returns a substring from a text string, starting at a specified position and with a specified length.
LEN	LEN(<text>)	Returns the number of characters in a text string.
UPPER	UPPER(<text>)	Converts a text string to uppercase.
LOWER	LOWER(<text>)	Converts a text string to lowercase.
TRIM	TRIM(<text>)	Removes leading and trailing spaces from a text string.

REPLACE	REPLACE(<text>, <start_position>, <num_chars>, <new_text>)	Replaces part of a text string with another text string, starting from a specified position.
SEARCH	SEARCH(<find_text>, <within_text>, <start_position>)	Returns the position of the first occurrence of a substring within another string.
EXACT	EXACT(<text1>, <text2>)	Compares two text strings and returns TRUE if they are exactly the same, otherwise returns FALSE.
TEXT	TEXT(<value>, <format>)	Converts a value to text according to the specified format.
VALUE	VALUE(<text>)	Converts a text string that represents a number to a numeric value.
SUBSTITUTE	SUBSTITUTE(<text>, <old_text>, <new_text>, <instance_num>)	Replaces occurrences of a substring with another substring.
TEXTJOIN	TEXTJOIN(<delimiter>, <ignore_empty>, <text1>, <text2>, ...)	Joins multiple text strings into one, with an optional delimiter and the option to ignore empty values.
FIND	FIND(<find_text>, <within_text>, <start_position>)	Returns the position of the first occurrence of a substring within another string, case-sensitive.

UNICHAR	UNICHAR(<unicode>)	Returns the Unicode character that corresponds to a specified code point.
UNICODE	UNICODE(<text>)	Returns the Unicode value of the first character in a text string.

8. Parent-Child Functions

DAX Formula Name	Syntax	Definition
PATH	PATH(<parent>, <child>)	Returns a delimited text string with the path from the parent to the child in a parent-child hierarchy.
PATHITEM	PATHITEM(<path>, <item_number>)	Returns a specific item from a path. The path is defined by the PATH function.
PATHITEMREVERSE	PATHITEMREVERSE(<path>, <item_number>)	Returns a specific item from a path in reverse order.
PATHLENGTH	PATHLENGTH(<path>)	Returns the length (number of items) of a path.
LOOKUPVALUE	LOOKUPVALUE(<result_column>, <search_column>, <search_value>)	Returns the value in a result column for a row where a specified column has a matching value.

9. Rank Functions

DAX Formula Name	Syntax	Definition
RANKX	RANKX(<table>, <expression>, <value>, <order>, <ties>)	Ranks the rows in a table based on the result of an expression, with the option to handle sorting and ties.
RANK.EQ	RANK.EQ(<number>, <values>, <order>)	Returns the rank of a number in a list of numbers, where duplicates receive the same rank.
RANK.AVG	RANK.AVG(<number>, <values>, <order>)	Returns the average rank of a number in a list of numbers, where duplicates receive the average rank.

10. Mathematical Functions

DAX Formula Name	Syntax	Definition
ABS	ABS(<number>)	Returns the absolute value of a number.
CEILING	CEILING(<number>, <significance>)	Rounds a number up, away from zero, to the nearest multiple of significance.
FLOOR	FLOOR(<number>, <significance>)	Rounds a number down, towards zero, to the nearest multiple of significance.
MOD	MOD(<number>, <divisor>)	Returns the remainder after a number is divided by a divisor.

POWER	POWER(<number>, <exponent>)	Returns the result of a number raised to the power of an exponent.
ROUND	ROUND(<number>, <num_digits>)	Rounds a number to a specified number of digits.
ROUNDUP	ROUNDUP(<number>, <num_digits>)	Rounds a number up, away from zero, to the specified number of digits.
ROUNDDOWN	ROUNDDOWN(<number>, <num_digits>)	Rounds a number down, towards zero, to the specified number of digits.
SQRT	SQRT(<number>)	Returns the square root of a number.
TRUNC	TRUNC(<number>, <num_digits>)	Truncates a number to a specified number of digits by removing the fractional part.
EXP	EXP(<number>)	Returns the constant e raised to the power of a number.
LN	LN(<number>)	Returns the natural logarithm of a number.
LOG	LOG(<number>, <base>)	Returns the logarithm of a number to the specified base.
PI	PI()	Returns the value of the mathematical constant pi.
RAND	RAND()	Returns a random number between 0 and 1.

SIGN	SIGN(<number>)	Returns the sign of a number, indicating whether the number is positive, negative, or zero.
SQUAREROOT	SQUAREROOT(<number>)	Returns the square root of a number.
LN10	LN10()	Returns the natural logarithm of 10.

11. Financial Functions

DAX Formula Name	Syntax	Definition
FV	FV(<rate>, <nper>, <pmt>, <pv>, <type>)	Returns the future value of an investment based on constant periodic payments and a constant interest rate.
IPMT	IPMT(<rate>, <period>, <nper>, <pv>, <fv>, <type>)	Returns the interest payment for an investment based on constant periodic payments and a constant interest rate.
PMT	PMT(<rate>, <nper>, <pv>, <fv>, <type>)	Returns the periodic payment for an investment based on constant periodic payments and a constant interest rate.
PV	PV(<rate>, <nper>, <pmt>, <fv>, <type>)	Returns the present value of an investment based on constant periodic payments and a constant interest rate.

RATE	RATE(<nper>, <pmt>, <pv>, <fv>, <type>, <guess>)	Returns the interest rate for an investment based on constant periodic payments and a constant present value.
NPER	NPER(<rate>, <pmt>, <pv>, <fv>, <type>)	Returns the number of periods for an investment based on constant periodic payments and a constant interest rate.

12. Miscellaneous Functions

DAX Formula Name	Syntax	Definition
BLANK	BLANK()	Returns a blank value.
DISTINCT	DISTINCT(<table>)	Returns a table with distinct values from a specified column.
EARLIER	EARLIER(<expression>, <level>)	Returns the value of an expression evaluated in an earlier row context in a nested iteration.
EARLIEST	EARLIEST(<expression>, <level>)	Returns the earliest value in a column for the current row context.
NEXTMONTH	NEXTMONTH(<dates>)	Returns the next month from the given date range.
PREVIOUSMONTH	PREVIOUSMONTH(<dates>)	Returns the previous month from the given date range.

USERELATIONSHIP	USERELATIONSHIP(<column1>, <column2>)	Specifies an alternative relationship to be used in the calculation.
EXACT	EXACT(<text1>, <text2>)	Compares two text strings and returns TRUE if they are exactly the same, otherwise returns FALSE.

13. Conversion Functions

DAX Formula Name	Syntax	Definition
VALUE	VALUE(<text>)	Converts a text string that represents a number to a numeric value.
DATEVALUE	DATEVALUE(<text>)	Converts a date in the form of text to a date/time value.
TIMEVALUE	TIMEVALUE(<text>)	Converts a time in the form of text to a time value.
TEXT	TEXT(<value>, <format>)	Converts a value to text according to the specified format.
INT	INT(<value>)	Converts a number to an integer by rounding down.
REAL	REAL(<value>)	Converts a text or number to a real number (float).
BOOLEAN	BOOLEAN(<value>)	Converts a value to a boolean (TRUE or FALSE).

14. Geospatial Functions

DAX Formula Name	Syntax	Definition
GEOADD	GEOADD(<geometry1>, <geometry2>)	Returns the geometry resulting from the union of two geometries, effectively combining them.
GEOMETRY	GEOMETRY(<geospatial_ expression>)	Returns the geometry of a geospatial object, enabling the handling and manipulation of spatial data.

Advanced Functions

DAX Formula Name	Syntax	Definition
CROSSJOIN	CROSSJOIN(<table1>, <table2>, ...)	Returns the Cartesian product of two or more tables. Every row from the first table is combined with every row from the second table.
UNION	UNION(<table1>, <table2>, ...)	Combines the rows of two or more tables, removing duplicates.
INTERSECT	INTERSECT(<table1>, <table2>)	Returns a table with the rows common to both tables.
EXCEPT	EXCEPT(<table1>, <table2>)	Returns a table with the rows that exist in the first table but not in the second table.

NATURALINNERJOIN	NATURALINNERJOIN(<table1>, <table2>)	Returns a table that contains only the rows where there is a match between both tables, joining on columns with the same names.
NATURALLEFTOUTERJOIN	NATURALLEFTOUTERJOIN(<table1>, <table2>)	Returns a table that contains all rows from the first table and matching rows from the second table, joining on columns with the same names.
GENERATE	GENERATE(<table1>, <expression>)	Returns a table by applying an expression to each row of the first table. The expression must return a table.
GENERATEALL	GENERATEALL(<table1>, <expression>)	Similar to GENERATE, but it returns all rows from the second table regardless of the filter context.
SUMMARIZE	SUMMARIZE(<table>, <group_by_column1>, <group_by_column2>, ...)	Returns a table grouped by one or more columns, with aggregated results for each group.
SUMMARIZECOLUMNS	SUMMARIZECOLUMNS(<column1>, <column2>, ...)	Returns a table with one or more columns of data, performing aggregations over the provided columns and expressions.

ADDCOLUMNS	ADDCOLUMNS(<table>, <new_column1>, <expression1>, ...)	Adds one or more columns to a table, evaluating the specified expressions for each row in the table.
SELECTCOLUMNS	SELECTCOLUMNS(<table>, <column1>, <expression1>, ...)	Returns a table with the specified columns, evaluating expressions for each column.


Debabrata Palit
/debabrata-palit03

DAX Mistakes That You Should Avoid

DAX ≠ Excel Functions

- DAX may look like Excel, but it behaves very differently—especially in how it handles filters and context.
- Example: *Total Sales = SUM(Sales[Amount])*
  - This looks like a SUM formula in Excel, but in DAX, it works within the filter context of the visual.
- Avoid it by understanding row context vs. filter context is key. Learn how Power BI applies filters to each visual and calculation.

Inconsistent Formatting and Referencing

- Inconsistent formatting and referencing makes DAX harder to read and debug.
- Do this instead: (1) Always prefix columns with their table name (e.g., Sales[Amount]). (2) Refer to measures directly without a table name (e.g., [Total Sales]). (3) Add comments for clarity.
- Example:

```
// Measure to calculate commission
Commission = [Total Sales] * 0.10
```

Neglecting Formatting and Comments

- Messy code leads to confusion and errors—especially in large models.
- Tip:
  - Use proper indentation.
  - Add comments using // to explain logic.
  - Group related measures logically.
- Example:


```
Total Sales = SUMX(Sales, Sales[Units] * RELATED(Products[Price]))
```

```
// Measure to calculate total sales from units and price
Total Sales =
    SUMX(
        Sales,
        Sales[Units] * RELATED(Products[Price])
    )
```

Misusing Scalar vs. Table Functions

- DAX functions are either scalar (return single values) or table functions (return tables), and mixing them improperly causes errors.
- Common mistake: *SUMX([Total Sales], [Commission])*
  - *SUMX expects a table as the first argument, not a measure.*

  Correct: *SUMX(Sales, Sales[Units] * RELATED(Products[Price]))*

Thinking Measures Are Like Table Rows

- Measures don’t operate row by row by default—you need to create a row context using iterators like SUMX, FILTER, or ADDColumns.

❌
```
Measure = IF(Sales[Price] > 20, "High", "Low")
```

✅
```
High Price Sales =
CALCULATE(
    [Total Sales],
    FILTER(Sales, Sales[Price] > 20)
)
```

THANK YOU!!!

STAY CONNECTED

SHARE YOUR THOUGHTS

SAVE FOR LATER

LIKE THIS POST