Most Common DAX Functions – Student Notes

This document includes common functions, what they do, and a simple example. students are expected to try these DAX functions for their dataset.

---

1) Aggregation Functions (Basic totals)

Function	What it does	Typical use	Example
`SUM()`	Adds values in a column	Total sales/revenue	`SUM(Sales[Amount])`
`AVERAGE()`	Average of values	Avg price, avg score	`AVERAGE(Sales[Amount])`
`MIN()`	Smallest value	Earliest date, lowest price	`MIN(Sales[Amount])`
`MAX()`	Largest value	Latest date, highest sales	`MAX(Sales[Amount])`
`COUNT()`	Counts non-blank values	Count orders/IDs	`COUNT(Sales[OrderID])`
`COUNTA()`	Counts non-empty values	Count product names	`COUNTA(Product[Name])`
`COUNTROWS()`	Counts rows in a table	Rows after filters	`COUNTROWS(Sales)`

---

2) Iterator Functions (Row-by-row calculations)

Function	What it does	Typical use	Example
`SUMX()`	Sums an expression evaluated per row	Qty × Price totals	`SUMX(Sales, Sales[Qty]*Sales[Price])`
`AVERAGEX()`	Averages an expression per row	Avg sales per order	`AVERAGEX(Sales, Sales[Amount])`
`MAXX()`	Max of an expression per row	Highest calculated value	`MAXX(Sales, Sales[Amount])`
`MINX()`	Min of an expression per row	Lowest calculated value	`MINX(Sales, Sales[Amount])`

Rule: If you need to calculate something per row first, use the **X** functions.

3) Filter & Context Functions (Core DAX)

Function	What it does	Typical use	Example
`CALCULATE()`	Changes filter context then evaluates expression	KPIs, conditional totals	`CALCULATE([Sales], Sales[Product]="Laptop")`
`FILTER()`	Returns a filtered table based on a condition	Complex filters	`FILTER(Sales, Sales[Qty] > 5)`
`ALL()`	Removes filters from a table/ column	% of total	`CALCULATE([Sales], ALL(Sales))`
`ALLEXCEPT()`	Removes filters except specified columns	Subtotals by a category	`ALLEXCEPT(Sales, Sales[Region])`
`REMOVEFILTERS()`	Clears filters (modern alternative to ALL in many cases)	Cleaner filter removal	`CALCULATE([Sales], REMOVEFILTERS(Sales))`
`KEEPFILTERS()`	Adds filters without replacing existing ones	Advanced filtering	`CALCULATE([Sales], KEEPFILTERS(Sales[Year]=2024))`

4) Logical Functions

Function	What it does	Typical use	Example
`IF()`	If/else condition	Labeling, rules	`IF([Sales] > 1000, "High", "Low")`
`SWITCH()`	Multiple conditions	Bucketing, bands	`SWITCH(TRUE(), [Sales]>1000, "High", "Low")`
`AND()`	Logical AND	Combined conditions	`AND(A, B)`
`OR()`	Logical OR	Alternative conditions	`OR(A, B)`

Function	What it does	Typical use	Example
`NOT()`	Logical NOT	Reverse logic	`NOT(A)`

5) Date & Time (Time Intelligence)

Best practice: Use a proper **Date table** (marked as Date table) for time intelligence.

Function	What it does	Typical use	Example
`YEAR()`	Extracts year	Year column/ logic	`YEAR(Sales[Date])`
`MONTH()`	Extracts month number	Month logic	`MONTH(Sales[Date])`
`TODAY()`	Current date	Current day KPIs	`TODAY()`
`TOTALYTD()`	Year-to-date total	YTD reports	`TOTALYTD([Sales], Date[Date])`
`TOTALMTD()`	Month-to-date total	MTD reports	`TOTALMTD([Sales], Date[Date])`
`SAMEPERIODLASTYEAR()`	Shifts to same period last year	YoY comparisons	`CALCULATE([Sales], SAMEPERIODLASTYEAR(Date[Date]))`
`DATEADD()`	Shifts dates by interval	Prior period	`CALCULATE([Sales], DATEADD(Date[Date], -1, YEAR))`

6) Text Functions

Function	What it does	Typical use	Example
`&`	Concatenates text	Labels	`Product[Name] & " - " & Product[Category]`
`CONCATENATE()`	Concatenates two strings	Simple join	`CONCATENATE(A, B)`

Function	What it does	Typical use	Example
`LEFT()`	Left N characters	Codes	`LEFT(Product[Code], 3)`
`RIGHT()`	Right N characters	IDs	`RIGHT(Product[ID], 2)`
`FORMAT()`	Formats a number/ date as text	Display formatting	`FORMAT([Sales], "#,##0")`

7) Math & Ranking

Function	What it does	Typical use	Example
`DIVIDE()`	Safe division (handles divide-by-zero)	Ratios, margins	`DIVIDE([Profit], [Sales])`
`ROUND()`	Rounds to N decimals	Reporting	`ROUND([Sales], 2)`
`ABS()`	Absolute value	Variance	`ABS([Variance])`
`RANKX()`	Ranks items based on a measure	Top N products	`RANKX(ALL(Product), [Sales])`

Common Measure Patterns (Must-know)

Pattern	What it's used for	Example
Total Sales	Base KPI	`Total Sales = SUM(Sales[Amount])`
% of Total	Contribution	`DIVIDE([Sales], CALCULATE([Sales], ALL(Sales)))`
Running Total	Cumulative trend	`CALCULATE([Sales], FILTER(ALL(Date), Date[Date] <= MAX(Date[Date])))`
YoY Difference	Year-over-year change	`[Sales] - CALCULATE([Sales], SAMEPERIODLASTYEAR(Date[Date]))`

Golden Rules for Students

- **Measures > Calculated Columns** for most analytics.
- `CALCULATE()` ** is the most important function** (it controls filter context).
- Use `FILTER()` for complex conditions.
- Use **X functions** for row-by-row calculations.
- DAX results change based on **filters, slicers, and visuals**.