DAX Library

📚 Pranjali’s DAX Template Library

🔝 1. TopN with Count

> 🧭 Use Case: Get top categories by total, e.g. top aircraft by crashes.

```
Top Category =
VAR RankedTable =
    ADDCOLUMNS(
        SUMMARIZE('Table', [Category]),
        "Total", SUM('Table'[Value])
    )
VAR TopRows =
    TOPN(3, RankedTable, [Total], DESC)
RETURN
    CONCATENATEX(
        TopRows,
        [Category] & " – " & [Total],
        ", "
    )
```

🔁 Replace `[Category]`, `[Value]`, and `3` as needed.

📅 2. Running Total (Year-To-Date)

> 🧭 Use Case: Track cumulative totals over time.

```
YTD Crashes =
CALCULATE(
    SUM('Table'[Crash Count]),
    DATESYTD('Table'[Date])
)
```

📌 Works best with a proper date table.

🧮 3. Dynamic Filtering Using SELECTEDVALUE + SWITCH

> 🧭 Use Case: Show value based on selected filter (like slicer-driven metric).

```
Dynamic Metric =
SWITCH(
    SELECTEDVALUE('Metric Selector'[Metric]),
    "Crashes", SUM('Table'[Crash Count]),
    "Fatalities", SUM('Table'[Fatalities]),
    "Survivors", SUM('Table'[Survivors]),
    BLANK()
)
```

🔁 You create a disconnected table with options: Crashes, Fatalities, Survivors.

🔀 4. Conditional Logic Using SWITCH

> 🧭 Use Case: Categorize rows into buckets.

```
Crash Severity =
SWITCH(
    TRUE(),
    'Table'[Fatalities] = 0, "Safe",
    'Table'[Fatalities] < 5, "Minor",
    'Table'[Fatalities] < 50, "Moderate",
    "Severe"
)
```

✅ `SWITCH(TRUE())` replaces messy nested IFs.

💯 5. % of Total (Share of Overall)

> 🧭 Use Case: Show what % a category contributes to the whole.

```
% of Total Crashes =
DIVIDE(
    SUM('Table'[Crash Count]),
    CALCULATE(SUM('Table'[Crash Count]), ALL('Table'))
)
```

📊 Useful in bar, pie, and tooltip visuals.

🔢 6. Rank by Measure

> 🧭 Use Case: Show ranking dynamically (e.g., top ranked aircraft by fatalities).

```
Aircraft Rank =
RANKX(
   ALL('Table'[Aircraft Type]),
   CALCULATE(SUM('Table'[Fatalities]))
)
```

🧭 Can be filtered by year, operator, etc.

⏱️ 7. YOY (Year-over-Year) % Change

> 🧭 Use Case: Compare metrics from this year to the previous year

```
YOY Crash % Change =
VAR CurrentYear = SUM('Table'[Crash Count])
VAR PreviousYear =
   CALCULATE(
      SUM('Table'[Crash Count]),
      SAMEPERIODLASTYEAR('Date'[Date])
   )
RETURN
   DIVIDE(CurrentYear - PreviousYear, PreviousYear)
```

✅ Must have a proper **Date table** marked as a date table.

📆 8. Month-over-Month (MoM) % Change

```
MoM Fatality % Change =
VAR ThisMonth = SUM('Table'[Fatalities])
VAR LastMonth =
    CALCULATE(
        SUM('Table'[Fatalities]),
        PARALLELPERIOD('Date'[Date], -1, MONTH)
    )
RETURN
    DIVIDE(ThisMonth - LastMonth, LastMonth)
```

🕒 9. Previous Value Lookup (e.g., Previous Month Crash Count)

```
Previous Month Crashes =
CALCULATE(
    SUM('Table'[Crash Count]),
    PREVIOUSMONTH('Date'[Date])
)
```

📊 10. Count of Unique Items

> 🧭 Use Case: How many unique aircraft types?

```
Unique Aircraft Types =
DISTINCTCOUNT('Table'[Aircraft Type])
```

🧠 11. Rolling 3-Month Total

> 🧭 Use Case: Smooth trends by summing recent 3 months

```
3M Rolling Fatalities =
CALCULATE(
    SUM('Table'[Fatalities]),
    DATESINPERIOD('Date'[Date], MAX('Date'[Date]), -3, MONTH)
)
```

🧮 12. Cumulative Count of Rows (Running Total of Incidents)

```
Cumulative Crashes =
CALCULATE(
    COUNTROWS('Table'),
    FILTER(
        ALLSELECTED('Table'),
        'Table'[Date] <= MAX('Table'[Date])
    )
)
```

🔗 13. Bridge Logic – Selected vs All

> 🧭 Use Case: Compare a selected value against the overall average

```
Crash vs Avg =
SUM('Table'[Crash Count])
    - CALCULATE(
        AVERAGE('Table'[Crash Count]),
        ALL('Table')
    )
```

🧾 14. Dynamic Title Based on Selection

> 🧭 Use Case: Change title as slicer selection changes

```
Title Measure =
"Crash Trend for: " &
IF(
    ISFILTERED('Table'[Operator]),
    SELECTEDVALUE('Table'[Operator]),
    "All Operators"
)
```

🚦 15. Status Flags / KPI Logic

> 🧭 Use Case: Create color-coded KPI status labels

```
KPI Status =
SWITCH(
    TRUE(),
    [YOY Crash % Change] > 0.1, "🚨 High Increase",
    [YOY Crash % Change] < -0.1, "✅ Strong Decrease",
    "⚠️ Moderate Change"
)
```

🔷 16. Logical & Information Functions

✅ `IFERROR` – Handle errors cleanly

```
Fatality Rate =
IFERROR(
    DIVIDE(SUM('Table'[Fatalities]), SUM('Table'[Aboard])),
    0
)
```

✅ `ISBLANK` – Check for missing values

```
dax
Has Fatalities =
IF(ISBLANK([Fatalities]), "Missing", "Recorded")
```

✅ `HASONEVALUE` – Useful in dynamic titles / filters

```
dax
Display Country =
IF(
    HASONEVALUE('Table'[Country]),
    VALUES('Table'[Country]),
    "Multiple Countries"
)
```

🔷 17. Text Functions

✅ CONCATENATE, CONCATENATEX, LEFT, RIGHT, MID, UPPER, LOWER

```
Formatted Aircraft =
CONCATENATE(
    UPPER(LEFT([Aircraft Type], 1)),
    LOWER(MID([Aircraft Type], 2, LEN([Aircraft Type])))
)
```

✅ SEARCH + CONTAINSSTRING – Smart filtering

```
IsMilitaryCrash =
CONTAINSSTRING([Summary], "military")
```

🔷 18. Date Functions (More Advanced)

✅ DATEDIFF – Time difference

```
CrashAge =
DATEDIFF([Manufacture Date], [Crash Date], YEAR)
```

✅ EOMONTH, STARTOFYEAR, ENDOFYEAR

```
CrashMonthEnd = EOMONTH([Date], 0)
```

🔷 19. Time Intelligence (Advanced)

✅ PARALLELPERIOD, PREVIOUSMONTH, NEXTMONTH

```
Previous Month Crashes =
CALCULATE(
    [Crash Count],
    PREVIOUSMONTH('Date'[Date])
)
```

🔷 20. Advanced Iterator Functions

✅ SUMX, AVERAGEX, FILTER

```
Weighted Fatality Rate =
AVERAGEX(
   VALUES('Table'[Aircraft Type]),
   DIVIDE(SUM('Table'[Fatalities]), SUM('Table'[Aboard]))
)
```

🔷 21. Ranking + Group Analysis

✅ RANKX, ALLSELECTED, REMOVEFILTERS

```
Crash Rank by Country =
RANKX(
   ALLSELECTED('Table'[Country]),
   [Crash Count]
)
```

🔷 22. Evaluation & Context Control

✅ SELECTEDVALUE, ISINSCOPE, REMOVEFILTERS

```
Selected Year = SELECTEDVALUE('Date'[Year])
```

```
Adjusted Crashes =
CALCULATE([Crash Count], REMOVEFILTERS('Table'[Operator]))
```

🔷 23. Debugging Tools (Helpful in Interview/Testing)

✅ VAR + RETURN – For cleaner logic and readability

```
Top Operator Crash Detail =
VAR Crashes = SUM('Table'[Crash Count])
VAR Operator = SELECTEDVALUE('Table'[Operator])
RETURN
Operator & ": " & Crashes
```