==
CATEGORY 2: FILTERING & LOGIC FUNCTIONS (12 Functions)
==

1️⃣1️⃣ FILTER() - FILTER TABLE BASED ON CONDITION

SYNTAX:
  FILTER(<table>, <condition>)

PARAMETERS:
  • <table>: Table to filter
  • <condition>: Boolean expression that must be TRUE

EXAMPLE FROM YOUR DATASET:

High Value Orders (>₹50,000):
  High Value Orders = FILTER(SalesData, SalesData[FinalAmount] > 50000)

Electronics Orders:
  Electronics Orders = FILTER(SalesData, SalesData[Category] = "Electronics")

North Region Orders:
  North Orders = FILTER(SalesData, SalesData[Region] = "North")

Combined Filter (AND condition):
  High Value Electronics = FILTER(SalesData,
    SalesData[FinalAmount] > 50000 && SalesData[Category] = "Electronics")

REAL USAGE:
  • Get filtered table for further calculations
  • Combine with SUM/AVERAGE for conditional aggregation
  • Create dynamic filtered lists

1️⃣2️⃣ CALCULATE() - CALCULATE WITH DIFFERENT CONTEXT

SYNTAX:
  CALCULATE(<expression>, <filter1>, <filter2>, ...)

PARAMETERS:
  • <expression>: Measure to calculate
  • <filter1>, <filter2>: Filters to apply to expression

EXAMPLE FROM YOUR DATASET:

Total Sales in North Region:
  North Sales = CALCULATE(SUM(SalesData[FinalAmount]), SalesData[Region] =
"North")
  Result: ~₹5,973,000

Total Electronics Sales:
  Electronics Sales = CALCULATE(SUM(SalesData[FinalAmount]), SalesData[Category]
= "Electronics")
  Result: ~₹8,900,000

Completed Orders Total:
  Completed Sales = CALCULATE(SUM(SalesData[FinalAmount]), SalesData[Status] =
"Completed")
  Result: ~₹25,100,000 (70% of all sales)

Multiple Filters:

```
  North Electronics = CALCULATE(SUM(SalesData[FinalAmount]),
    SalesData[Region] = "North",
    SalesData[Category] = "Electronics")
  Result: Specific to North region electronics

REAL USAGE:
  • Most important DAX function!
  • Calculate with specific criteria
  • Change filter context dynamically
  • Build complex business rules
```

1️⃣3️⃣ IF() - CONDITIONAL LOGIC

```
SYNTAX:
  IF(<condition>, <true_value>, <false_value>)

PARAMETERS:
  • <condition>: Boolean condition to test
  • <true_value>: Result if TRUE
  • <false_value>: Result if FALSE

EXAMPLE FROM YOUR DATASET:

Simple IF in Calculated Column:
  Order Priority = IF(SalesData[FinalAmount] > 50000, "High Priority", "Low
Priority")

Nested IF (Calculated Column):
  Order Category =
    IF(SalesData[FinalAmount] >= 50000, "Premium",
    IF(SalesData[FinalAmount] >= 20000, "Standard",
    IF(SalesData[FinalAmount] >= 10000, "Economy", "Budget")))

  Result:
    • ₹50,000+: Premium
    • ₹20,000-₹50,000: Standard
    • ₹10,000-₹20,000: Economy
    • <₹10,000: Budget

Status Classification (Calculated Measure):
  Sales Status = IF(SUM(SalesData[Amount]) > 100000, "Target Met", "Below
Target")

REAL USAGE:
  • Create categories/segments
  • Conditional calculations
  • Business rule implementation
  • Risk assessments
```

1️⃣4️⃣ SWITCH() - MULTIPLE CONDITIONS (CLEANER THAN NESTED IF)

```
SYNTAX:
  SWITCH(<expression>, <value1>, <result1>, <value2>, <result2>, ..., <default>)

PARAMETERS:
  • <expression>: Value to compare
  • <value1>, <value2>: Values to match
  • <result1>, <result2>: Results if match
```

```
  • <default>: Result if no match

EXAMPLE FROM YOUR DATASET:

Region Bonus Percentage (Calculated Column):
  Regional Bonus = SWITCH(SalesData[Region],
    "North", 10,
    "South", 12,
    "East", 8,
    "West", 15,
    "Central", 9,
    "Northeast", 7,
    0)

  Result:
    • North region: 10% bonus
    • West region: 15% bonus (highest)
    • Northeast: 7% bonus (lowest)

Category Commission Rate (Calculated Column):
  Commission = SWITCH(SalesData[Category],
    "Electronics", 5,
    "Clothing", 8,
    "Home & Kitchen", 6,
    "Books", 4,
    "Sports", 7,
    0)

REAL USAGE:
  • Cleaner than nested IF
  • Category-based calculations
  • Commission/bonus rates
  • Status classifications
```

1️⃣5️⃣ AND() - LOGICAL AND (ALL CONDITIONS TRUE)

```
SYNTAX:
  AND(<condition1>, <condition2>, ...)

PARAMETERS:
  • <condition1>, <condition2>: Boolean conditions

EXAMPLE FROM YOUR DATASET:

High Value Electronics Order (Calculated Column):
  Is High Value Electronics = IF(AND(
    SalesData[FinalAmount] > 50000,
    SalesData[Category] = "Electronics"),
    "Yes", "No")

  Result: "Yes" only if BOTH conditions are true:
    • Amount > ₹50,000 AND
    • Category = Electronics

Completed Premium Order:
  Is Priority = IF(AND(
    SalesData[Status] = "Completed",
    SalesData[FinalAmount] > 75000),
    "Priority", "Regular")

REAL USAGE:
```

- Multiple conditions must ALL be true
- Complex filtering logic
- Multi-criteria classification

---

16 OR() - LOGICAL OR (ANY CONDITION TRUE)

```
SYNTAX:
  OR(<condition1>, <condition2>, ...)

PARAMETERS:
  • <condition1>, <condition2>: Boolean conditions

EXAMPLE FROM YOUR DATASET:

High Value OR Clothing Order (Calculated Column):
  Is Target Order = IF(OR(
    SalesData[FinalAmount] > 75000,
    SalesData[Category] = "Clothing"),
    "Target", "Other")

  Result: "Target" if EITHER condition is true:
    • Amount > ₹75,000 OR
    • Category = Clothing

Pending OR Cancelled (Calculated Column):
  Need Action = IF(OR(
    SalesData[Status] = "Pending",
    SalesData[Status] = "Cancelled"),
    "Yes", "No")

REAL USAGE:
  • ANY condition can be true
  • Inclusive filtering
  • Multi-category selection
```

---

17 NOT() - LOGICAL NEGATION

```
SYNTAX:
  NOT(<condition>)

PARAMETERS:
  • <condition>: Boolean condition to negate

EXAMPLE FROM YOUR DATASET:

NOT Completed (Calculated Column):
  Not Delivered = IF(NOT(SalesData[Status] = "Completed"), "Yes", "No")

NOT Electronics (Calculated Column):
  Is Non Electronics = IF(NOT(SalesData[Category] = "Electronics"), "Yes", "No")

Invert High Value Logic:
  Is Not Premium = IF(NOT(SalesData[FinalAmount] > 50000), "Yes", "No")

REAL USAGE:
  • Negate conditions
  • Opposite logic
  • Exclude specific cases
```

1️⃣8️⃣ IFERROR() - ERROR HANDLING

```
SYNTAX:
  IFERROR(<expression>, <error_value>)

PARAMETERS:
  • <expression>: Expression that might error
  • <error_value>: Value if error occurs

EXAMPLE FROM YOUR DATASET:

Safe Division (Calculated Measure):
  Average Price = IFERROR(
    SUM(SalesData[Amount]) / COUNT(SalesData[OrderID]),
    0)

  Result: Shows 0 if there's a division error (rarely happens)

Safe LOOKUPS:
  Customer Lookup = IFERROR(
    RELATED(CustomerMaster[CustomerName]),
    "Unknown Customer")

REAL USAGE:
  • Prevent #ERROR displays
  • Handle null/blank values
  • Safe calculations
  • Production-ready formulas
```

1️⃣9️⃣ HASONEVALUE() - CHECK SINGLE VALUE CONTEXT

```
SYNTAX:
  HASONEVALUE(<column>)

PARAMETERS:
  • <column>: Column to check

EXAMPLE FROM YOUR DATASET:

Show Region Name Only If Single (Calculated Measure):
  Region Display = IF(HASONEVALUE(SalesData[Region]),
    VALUES(SalesData[Region]),
    "Multiple Regions")

  Result:
    • If filtering by one region: Shows "North"
    • If multiple regions selected: Shows "Multiple Regions"
    • If no filter: Shows "Multiple Regions"

Use in KPI:
  KPI Title = IF(HASONEVALUE(SalesData[Category]),
    "Category: " & VALUES(SalesData[Category]),
    "All Categories")

REAL USAGE:
  • Check filter context
  • Conditional text display
```

- Dynamic titles/labels

2️⃣0️⃣ ALL() - REMOVE ALL FILTERS

```
SYNTAX:
  ALL(<column>)
  ALL(<table>)

PARAMETERS:
  • <column>: Column to clear filters from
  • <table>: Table to clear filters from

EXAMPLE FROM YOUR DATASET:

Total Sales Regardless of Filters (Calculated Measure):
  Grand Total = CALCULATE(SUM(SalesData[FinalAmount]), ALL(SalesData))

  Result: Always shows ₹35,860,529 (total of ALL 1000 orders)
  Even if you filter by Region/Category - shows grand total

Total by Region (Calculated Measure):
  Region Total = CALCULATE(SUM(SalesData[FinalAmount]), ALL(SalesData[Region]))

  Result: Total across all regions (clears region filter)

REAL USAGE:
  • Calculate grand total
  • Percentage of grand total
  • Ratio calculations
  • Overall benchmarks
```

2️⃣1️⃣ ALLSELECTED() - CLEAR FILTERS BUT RESPECT VISUAL

```
SYNTAX:
  ALLSELECTED(<column>)
  ALLSELECTED(<table>)

PARAMETERS:
  • <column>: Column to clear filters from
  • <table>: Table to clear filters from

EXAMPLE FROM YOUR DATASET:

Visual Grand Total (Calculated Measure):
  Visual Total = CALCULATE(SUM(SalesData[FinalAmount]), ALLSELECTED())

  Result:
    • In a table filtered to "Electronics": Shows total of only Electronics
    • Clears other filters but respects the visual's context

Percentage of Visual Total:
  % of Visual Total = DIVIDE(
    SUM(SalesData[FinalAmount]),
    CALCULATE(SUM(SalesData[FinalAmount]), ALLSELECTED()),
    0)

REAL USAGE:
  • Visual-specific grand totals
```

- Percentage within visual
- Row-level percentages

2️⃣2️⃣ ISBLANK() - CHECK IF VALUE IS BLANK

```
SYNTAX:
  ISBLANK(<expression>)

PARAMETERS:
  • <expression>: Value to check

EXAMPLE FROM YOUR DATASET:

Check Empty Email (Calculated Column):
  Has Email = IF(ISBLANK(CustomerMaster[Email]), "No Email", "Has Email")

Handle Missing Phone (Calculated Column):
  Phone Status = IF(ISBLANK(SalesData[DeliveryDate]),
    "Not Yet Delivered",
    "Delivered")

Safe Value Replacement:
  Customer Name Display = IF(ISBLANK(SalesData[CustomerID]),
    "Anonymous",
    RELATED(CustomerMaster[CustomerName]))

REAL USAGE:
  • Data quality checks
  • Handle missing values
  • Conditional display
```