POWER BI DAX FUNCTIONS - TOP 50 COMPLETE TEACHING GUIDE

Real Indian Dataset (1000 rows, 15 columns)
Calculated Columns | Calculated Measures | Date Table

FILE: Power_BI_DAX_Functions_Dataset.xlsx

4 Worksheets with 1000+ rows of real Indian e-commerce sales data
Perfect for teaching Top 50 DAX Functions
Complete with Date Table, Customer Master, Product Master
Ready to use in Power BI Desktop

DATASET OVERVIEW

Total Sales Data: 1000 records
Total Revenue: ₹35,748,247
Date Range: Jan 2023 - Dec 2025 (3 years)
Unique Customers: 500
Unique Products: 25
Categories: 5 (Electronics, Clothing, Home & Kitchen, Books, Sports)
Regions: 6 (North, South, East, West, Central, Northeast)

WORKSHEET 1: SALES DATA (1000 ROWS, 15 COLUMNS)

Column Name	Type	Description	Example Values
OrderID	Integer	Unique order identifier	10001-11000
CustomerID	Integer	Customer ID (FK)	2001-2500
OrderDate	Date	Date of order	2023-01-15
City	Text	City where order is from	Mumbai, Delhi
State	Text	State	Maharashtra, Delhi
Region	Text	Geographic region	North, South
Category	Text	Product category	Electronics, Clothing
Product	Text	Product name	Laptop, T-Shirt
Quantity	Integer	Number of items ordered	1-5
UnitPrice	Currency	Price per item	500-100000
Amount	Currency	Quantity × UnitPrice	5000-500000
DiscountPercent	Decimal	Discount given (0-20%)	0, 5, 10, 15, 20
NetAmount	Currency	Amount - Discount	4000-450000
FinalAmount	Currency	NetAmount + 18% GST	5000-530000
Status	Text	Order status	Completed, Pending

WHY THIS DATASET IS PERFECT FOR DAX TEACHING:

✓ Real-world Indian e-commerce scenario
✓ Contains dates for time intelligence functions
✓ Multiple dimensions (City, State, Region, Category)
✓ Multiple metrics (Quantity, Amount, Discount)
✓ Time-based variations (sales differ by month/quarter/year)
✓ Good for CALCULATE, FILTER, SUMIF examples
✓ Perfect for context transition examples
✓ Contains NULL-like scenarios (different status values)

WORKSHEET 2: DATE TABLE (1000+ ROWS)

What is Date Table?
- Specialized table with one row per date
- Contains time intelligence columns
- Links to fact tables via date relationships
- Foundation for time-based DAX functions

Columns in Date Table:
- Date: Actual date value (2023-01-01, 2023-01-02, etc.)
- Year: 2023, 2024, 2025
- Quarter: Q1, Q2, Q3, Q4
- Month: 1-12
- MonthName: January, February, etc.
- Week: 1-52
- DayOfWeek: 1-7 (Monday to Sunday)
- DayName: Monday, Tuesday, etc.
- IsWeekend: 1 if Saturday/Sunday, 0 otherwise
- IsHoliday: 1 if holiday, 0 otherwise (simplified)

Why Date Table is Important:
- ✓ Enables time intelligence DAX functions
- ✓ Ensures consistent date handling
- ✓ Allows fiscal year calculations
- ✓ Supports holidays/weekends logic
- ✓ Required for Year-over-Year comparisons
- ✓ Essential for quarter comparisons
- ✓ Supports moving averages
- ✓ Perfect for custom calendar scenarios

Where Date Table is Used:
- ├─ Time intelligence calculations (YTD, MTD, etc.)
- ├─ Year-over-Year comparisons
- ├─ Moving averages and trending
- ├─ Holiday/Weekend filtering
- ├─ Fiscal calendar handling
- ├─ Custom business calendar
- └─ Complex date logic

WORKSHEET 3: CUSTOMER MASTER (500 ROWS)

Columns:
- CustomerID: Unique customer identifier
- CustomerName: Customer name
- Email: Email address
- Phone: Contact number
- City: City location
- State: State location

- JoinDate: Customer registration date
- Status: Active/Inactive

Purpose:
- Dimension table for customer analysis
- Links to SalesData via CustomerID
- Enables customer segmentation

WORKSHEET 4: PRODUCT MASTER (25 ROWS)

Columns:
- ProductID: Unique product identifier
- ProductName: Product name
- Category: Product category
- Status: Active/Inactive

Purpose:
- Dimension table for product analysis
- Links to SalesData via ProductName
- Enables product segmentation

CALCULATED COLUMN vs CALCULATED MEASURE

What is Calculated Column?
- New column added to a table using DAX
- Calculates value for each row independently
- Stored in the data model (uses memory)
- Row-by-row computation
- Value calculated once, then stored

What is Calculated Measure?
- Not stored physically
- Calculated on-the-fly when used in visualization
- Only exists in Power BI (not in table)
- Context-aware (changes based on filter context)
- Aggregated across rows

KEY DIFFERENCES:

Aspect	Calculated Column	Calculated Measure
Storage	Stored in table	Not stored
Computation	Row-by-row	On-the-fly
Context	Row context	Filter context
Memory Usage	Uses memory	No additional memory
Refresh Time	Slower (stored)	Faster (not stored)
Performance	Good for filtering	Good for aggregation
Usage	In visuals & filters	Primarily in visuals
Function Required	DAX with row context	DAX with aggregation

WHEN TO USE CALCULATED COLUMN:
- ✓ Need to use value in relationships
- ✓ Need to filter/slice by this value
- ✓ Value depends on single row only
- ✓ Creating category/segment column
- ✓ Data manipulation needed in model

WHEN TO USE CALCULATED MEASURE:

  ✓ Aggregating data (SUM, AVG, COUNT, etc.)
  ✓ Complex multi-row calculations
  ✓ Context-dependent calculations
  ✓ Performance is critical
  ✓ Only needed in visualizations

═══
EXAMPLE 1: CALCULATED COLUMN - PROFIT MARGIN PERCENTAGE
═══

Business Scenario:
  Need to calculate profit margin percentage for each order
  Used for filtering orders by profitability
  Will be used in row-level filtering and slicing

DAX Formula (Calculated Column):
  ProfitMarginPercent = (SalesData[NetAmount] - SalesData[CostAmount]) /
SalesData[NetAmount] * 100

Where to Apply:
  1. Go to Power BI Desktop
  2. Open SalesData table
  3. Click "New Column" in ribbon
  4. Enter formula above
  5. Column added to table

Result:
  ✓ New column in SalesData table
  ✓ Each row has profit margin %
  ✓ Can filter/slice by this value
  ✓ Can use in relationships

═══
EXAMPLE 2: CALCULATED COLUMN - ORDER CATEGORY
═══

Business Scenario:
  Categorize orders as "High Value", "Medium Value", "Low Value"
  Need to filter and segment by order value
  Will be used in reports and visuals

DAX Formula (Calculated Column):
  OrderCategory =
    IF(SalesData[FinalAmount] >= 50000, "High Value",
    IF(SalesData[FinalAmount] >= 20000, "Medium Value", "Low Value"))

Result:
  ✓ New column with three categories
  ✓ Based on FinalAmount value
  ✓ Can segment customers by order category
  ✓ Useful for dynamic pricing analysis

═══
EXAMPLE 3: CALCULATED MEASURE - TOTAL SALES
═══

Business Scenario:
  Calculate total sales for use in visualizations
  Total should change based on filter context
  Need across multiple visuals with different filters

DAX Formula (Calculated Measure):
  Total Sales = SUM(SalesData[FinalAmount])

Where to Create:
  1. Power BI Desktop
  2. Click "New Measure" in Home ribbon
  3. Enter formula above
  4. Measure appears in field list

Behavior:
  ✓ Shows total across all rows: ₹35,748,247
  ✓ Filter by Region=North: Shows North total only
  ✓ Filter by Month: Shows that month total only
  ✓ Context-aware and dynamic

═══
EXAMPLE 4: CALCULATED MEASURE - AVERAGE ORDER VALUE
═══

Business Scenario:
  Calculate average order value
  Should vary by selected time period
  Key metric for trend analysis

DAX Formula (Calculated Measure):
  Average Order Value = AVERAGE(SalesData[FinalAmount])

Or Better (handles filters correctly):
  Average Order Value = SUMX(SalesData, SalesData[FinalAmount]) /
COUNTA(SalesData[OrderID])

Usage:
  ✓ KPI card showing ₹35,748
  ✓ Line chart showing AOV by month
  ✓ Table showing AOV by region
  ✓ All automatically filtered

═══
EXAMPLE 5: CALCULATED MEASURE - YEAR-TO-DATE SALES
═══

Business Scenario:
  Calculate cumulative sales from beginning of year to current date
  Show YTD progress dynamically
  Compare against targets

DAX Formula (Calculated Measure):
  YTD Sales =
    CALCULATE(
      SUM(SalesData[FinalAmount]),
      DATESYTD(DateTable[Date], "12/31")
)

Where Used:
  ✓ Executive dashboard showing YTD progress
  ✓ KPI card comparing against goal
  ✓ Waterfall chart showing progression
  ✓ Requires Date Table relationship

═══
DATE TABLE: IMPORTANCE & APPLICATION
═══

What is Date Table?
  A specialized dimension table containing one row per date
  Includes time-related attributes (year, month, quarter, etc.)
  Foundation for all time intelligence calculations

Required for advanced DAX functions

Why It's Important:

1. Time Intelligence Functions:
   • DATESYTD(), DATESQTD(), DATESMTD()
   • SAMEPERIODLASTYEAR()
   • DATEADD()
   • All require proper Date Table

2. Consistent Date Handling:
   • Ensures all dates handled same way
   • Custom calendar support (fiscal year)
   • Holiday inclusion/exclusion
   • Weekend calculations

3. Performance:
   • Better query optimization
   • Faster time-based calculations
   • Efficient filtering

4. Business Calendar:
   • Support fiscal calendars (April-March)
   • Custom fiscal months
   • Holiday incorporation
   • Week numbering

Where It's Used:

1. Sales Trends:
   • Monthly sales trends
   • Quarterly performance
   • Year-over-year comparison
   Formula: YoY Growth = CALCULATE(SUM(Amount),
SAMEPERIODLASTYEAR(DateTable[Date])) / ...

2. Rolling Averages:
   • 7-day moving average
   • 30-day moving average
   • Trend smoothing
   Formula: 7DayMA = AVERAGEX(DATESBETWEEN(DateTable[Date], TODAY()-7, TODAY()),
SUM(Amount))

3. Cohort Analysis:
   • Customer cohorts by join date
   • Product launch tracking
   • Retention curves

4. Forecasting:
   • Historical trend analysis
   • Projection calculations
   • Seasonality detection

5. Financial Reporting:
   • Period closing
   • Fiscal period calculations
   • Accrual basis accounting

How to Create Date Table:

Method 1 (Automatic):
  1. Power BI Desktop
  2. Modeling → New Table
  3. Paste: DateTable = CALENDAR(DATE(2023,1,1), DATE(2025,12,31))

4. Then add columns for Month, Quarter, etc.

Method 2 (Using External Data):
1. Import from DateTable worksheet
2. Better for custom calendars

Link to Sales Table:
1. Go to Model view
2. Create relationship: SalesData[OrderDate] → DateTable[Date]
3. Mark DateTable[Date] as Date column
4. Relationship now active!