# Power BI DAX Function Examples

## 1. FILTER Function Examples

The FILTER function returns a table with rows that meet specified conditions.

### Basic FILTER Example

dax

```dax
-- Get all sales where amount is greater than 1000
FilteredSales = FILTER(Sales, Sales[Amount] > 1000)
```

### FILTER with Multiple Conditions

dax

```dax
-- Filter sales for specific product and year
HighValueElectronics = FILTER(
    Sales,
    Sales[ProductCategory] = "Electronics" &&
    Sales[Amount] > 500 &&
    YEAR(Sales[OrderDate]) = 2023
)
```

### Using FILTER in Calculate

dax

```dax
-- Total sales for high-value transactions
HighValueSalesTotal = CALCULATE(
    SUM(Sales[Amount]),
    FILTER(Sales, Sales[Amount] > 1000)
)
```

### FILTER with Related Tables

dax

```dax
-- Sales from customers in specific city
CityBasedSales = CALCULATE(
    SUM(Sales[Amount]),
    FILTER(
        Sales,
        RELATED(Customer[City]) = "New York"
    )
)
```

### Complex FILTER with Variables

dax

```dax
RecentHighValueSales =
VAR RecentDate = TODAY() - 30
RETURN
CALCULATE(
    SUM(Sales[Amount]),
    FILTER(
        Sales,
        Sales[OrderDate] >= RecentDate &&
        Sales[Amount] > AVERAGE(Sales[Amount])
    )
)
```

## 2. Date and Time Functions

### DATE Functions

dax

```dax
-- Create a date from year, month, day
CreatedDate = DATE(2023, 12, 25)

-- Extract year, month, day
SalesYear = YEAR(Sales[OrderDate])
SalesMonth = MONTH(Sales[OrderDate])
SalesDay = DAY(Sales[OrderDate])

-- Get day of week
DayOfWeek = WEEKDAY(Sales[OrderDate], 2)  -- 2 = Monday as 1
DayName = FORMAT(Sales[OrderDate], "dddd")
```

### TIME Functions

dax

```dax
-- Create time
MeetingTime = TIME(14, 30, 0)  -- 2:30 PM

-- Extract time components
OrderHour = HOUR(Sales[OrderDateTime])
OrderMinute = MINUTE(Sales[OrderDateTime])
OrderSecond = SECOND(Sales[OrderDateTime])
```

## Date Calculations

dax

```dax
-- Add/subtract days
NextWeek = Sales[OrderDate] + 7
LastMonth = Sales[OrderDate] - 30

-- Date difference
DaysToShip = DATEDIFF(Sales[OrderDate], Sales[ShipDate], DAY)
MonthsBetween = DATEDIFF(Sales[OrderDate], TODAY(), MONTH)

-- End of period dates
EndOfMonth = EOMONTH(Sales[OrderDate], 0)
EndOfYear = DATE(YEAR(Sales[OrderDate]), 12, 31)
```

## Advanced Date Functions

dax

```dax
-- Current date and time
CurrentDate = TODAY()
CurrentDateTime = NOW()

-- Beginning of periods
StartOfMonth = DATE(YEAR(Sales[OrderDate]), MONTH(Sales[OrderDate]), 1)
StartOfYear = DATE(YEAR(Sales[OrderDate]), 1, 1)

-- Calendar functions
IsWeekend = WEEKDAY(Sales[OrderDate]) IN {1, 7}  -- Sunday=1, Saturday=7
QuarterNumber = ROUNDUP(MONTH(Sales[OrderDate]) / 3, 0)

-- Date formatting
FormattedDate = FORMAT(Sales[OrderDate], "MMM DD, YYYY")
ShortDate = FORMAT(Sales[OrderDate], "MM/DD/YY")
```

### Time Intelligence Examples

dax

```dax
-- Year-to-date sales
YTDSales = TOTALYTD(SUM(Sales[Amount]), Sales[OrderDate])

-- Previous month sales
PreviousMonthSales = CALCULATE(
    SUM(Sales[Amount]),
    PREVIOUSMONTH(Sales[OrderDate])
)

-- Same period last year
SamePeriodLastYear = CALCULATE(
    SUM(Sales[Amount]),
    SAMEPERIODLASTYEAR(Sales[OrderDate])
)
```

## 3. Logical Functions

### IF Function

dax

```dax
-- Simple IF
SalesCategory = IF(Sales[Amount] > 1000, "High", "Low")

-- Nested IF
SalesGrade = IF(
    Sales[Amount] > 5000, "A",
    IF(Sales[Amount] > 2000, "B",
    IF(Sales[Amount] > 500, "C", "D"))
)

-- IF with complex conditions
BonusEligible = IF(
    Sales[Amount] > 1000 && Sales[Customer] <> BLANK(),
    "Yes",
    "No"
)
```

### SWITCH Function

dax

```dax
-- SWITCH is cleaner than nested IFs
ProductGroup = SWITCH(
    Sales[ProductCategory],
    "Electronics", "Tech",
    "Clothing", "Fashion",
    "Books", "Media",
    "Food", "Consumables",
    "Other"
)

-- SWITCH with TRUE for conditions
PriceRange = SWITCH(
    TRUE(),
    Sales[Amount] >= 5000, "Premium",
    Sales[Amount] >= 1000, "Standard",
    Sales[Amount] >= 100, "Basic",
    "Economy"
)
```

## AND, OR, NOT Functions

dax

```dax
-- AND function
HighValueRecent = IF(
    AND(Sales[Amount] > 1000, Sales[OrderDate] > DATE(2023, 1, 1)),
    "Yes",
    "No"
)

-- OR function
SpecialCustomer = IF(
    OR(Sales[Amount] > 5000, Sales[CustomerType] = "VIP"),
    "Special",
    "Regular"
)

-- NOT function
NotElectronics = IF(
    NOT(Sales[ProductCategory] = "Electronics"),
    "Non-Tech",
    "Tech"
)
```

### IFERROR and ISBLANK

dax

```dax
-- Handle errors gracefully
SafeDivision = IFERROR(Sales[Amount] / Sales[Quantity], 0)

-- Check for blank values
HasCustomer = IF(ISBLANK(Sales[Customer]), "No Customer", "Has Customer")

-- Multiple blank checks
DataQuality = IF(
    OR(ISBLANK(Sales[Customer]), ISBLANK(Sales[Product])),
    "Incomplete",
    "Complete"
)
```

## 4. String Functions

### Basic String Functions

dax

```dax
-- String length
CustomerNameLength = LEN(Customer[CustomerName])

-- Upper and lower case
UpperName = UPPER(Customer[CustomerName])
LowerName = LOWER(Customer[CustomerName])
ProperName = PROPER(Customer[CustomerName])
```

### String Extraction

dax

```dax
-- LEFT, RIGHT, MID functions
FirstThreeChars = LEFT(Product[ProductCode], 3)
LastFourChars = RIGHT(Product[ProductCode], 4)
MiddleChars = MID(Product[ProductCode], 3, 4)  -- Start at 3, take 4 chars

-- FIND position of substring
AtPosition = FIND("@", Customer[Email])
DomainStart = FIND("@", Customer[Email]) + 1
```

### String Manipulation

dax

```dax
-- Concatenation
FullName = Customer[FirstName] & " " & Customer[LastName]
-- Alternative concatenation
FullNameAlt = CONCATENATE(Customer[FirstName], CONCATENATE(" ", Customer[LastName]))

-- SUBSTITUTE (replace text)
CleanedPhone = SUBSTITUTE(SUBSTITUTE(Customer[Phone], "(", ""), ")", "")
StandardizedCode = SUBSTITUTE(Product[ProductCode], "-", "_")

-- TRIM (remove spaces)
CleanName = TRIM(Customer[CustomerName])
```

## Advanced String Functions

dax

```dax
-- Extract email domain
EmailDomain = RIGHT(
    Customer[Email],
    LEN(Customer[Email]) - FIND("@", Customer[Email])
)

-- Create initials
CustomerInitials = LEFT(Customer[FirstName], 1) & LEFT(Customer[LastName], 1)

-- Format phone numbers
FormattedPhone =
VAR CleanPhone = SUBSTITUTE(SUBSTITUTE(SUBSTITUTE(Customer[Phone], "(", ""), ")", ""), "-", "")
RETURN
IF(
    LEN(CleanPhone) = 10,
    "(" & LEFT(CleanPhone, 3) & ") " & MID(CleanPhone, 4, 3) & "-" & RIGHT(CleanPhone, 4),
    Customer[Phone]
)

-- Extract first word
FirstWord = LEFT(Product[ProductName], FIND(" ", Product[ProductName] & " ") - 1)

-- Check if string contains substring
ContainsWord = IF(FIND("Premium", Product[ProductName], 1, 0) > 0, "Yes", "No")
-- Alternative using SEARCH (case-insensitive)
ContainsWordIgnoreCase = IF(SEARCH("premium", Product[ProductName], 1, 0) > 0, "Yes", "No")
```

## String Formatting

dax

```dax
-- FORMAT function for strings
FormattedAmount = FORMAT(Sales[Amount], "$#,##0.00")
FormattedPercent = FORMAT(Sales[DiscountRate], "0.00%")
PaddedID = FORMAT(Customer[CustomerID], "000000")  -- Pad with zeros

-- Combine string functions for complex formatting
ProductDisplay = UPPER(LEFT(Product[Category], 1)) &
                 LOWER(RIGHT(Product[Category], LEN(Product[Category]) - 1)) &
                 " - " &
                 PROPER(Product[ProductName])
```

## Practical String Examples

dax

```dax
-- Create URL-friendly slugs
URLSlug = SUBSTITUTE(SUBSTITUTE(LOWER(Product[ProductName]), " ", "-"), "&", "and")

-- Mask sensitive data
MaskedSSN = LEFT(Customer[SSN], 3) & "-XX-" & RIGHT(Customer[SSN], 4)

-- Extract file extension
FileExtension = RIGHT(Document[FileName], LEN(Document[FileName]) - FIND(".", Document[FileName

-- Validate email format
IsValidEmail = IF(
    AND(
        FIND("@", Customer[Email], 1, 0) > 0,
        FIND(".", Customer[Email], FIND("@", Customer[Email]), 0) > 0
    ),
    "Valid",
    "Invalid"
)
```