# Power BI - Advanced DAX - p3 - Lecture 22

## ASSIGNMENT: CALCULATE & ALL

1.Create a new measure named All Returns to calculate the total number of returns, regardless of filter context


```
All Returns = CALCULATE([Total Returns],ALL('Returns Data'))
```

2. Create a new measure named % of All Returns that divides Total Returns by All Returns

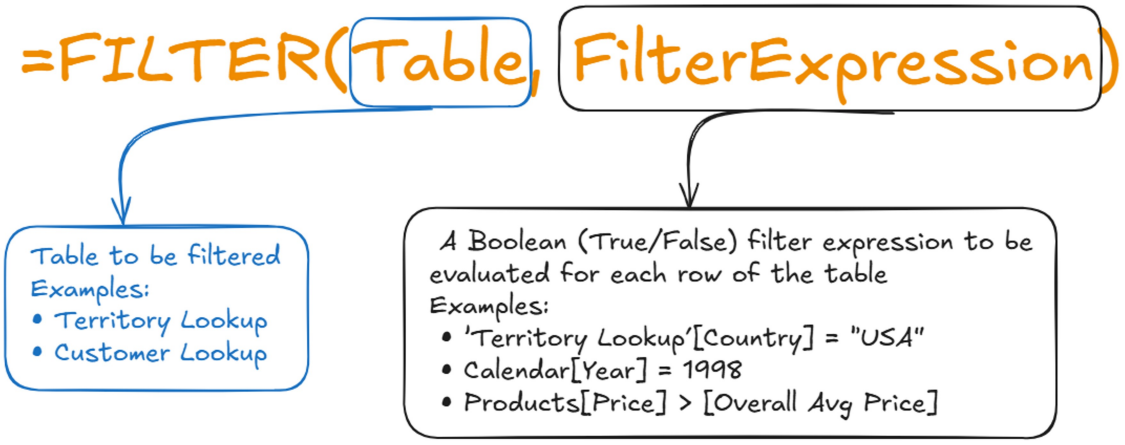
```
% of All Returns = DIVIDE([Total Returns],[All Returns])
```

3.Create a matrix to show % of All Returns (values) by product Category Name (rows). Which category accounts for the largest percentage of returns? The smallest?

| CategoryName | Total Returns | All Returns | % of All Returns | |
|---|---|---|---|---|
| Accessories | 1130 | 1828 | 61.82% | largest |
| Bikes | 429 | 1828 | 23.47% | |
| Clothing | 269 | 1828 | 14.72% | smallest |
| Components | | 1828 | | |
| **Total** | **1828** | **1828** | **100.00%** | |

## FILTER

FILTER() :-
Returns a table that represents a subset of another table or expression



HEY THIS IS IMPORTANT!

- FILTER is used to add new filter context, and can handle more complex filter expressions than CALCULATE (by referencing measures, for example)

- Since FILTER returns an entire table, it's often nested within other functions, like CALCULATE or SUMX

Calculate(Expression,FILTER(Table, Advance Filters))

PRO TIP:
- Since FILTER iterates through each row in a table, it can be slow and computationally expensive; only use FILTER if a simple CALCULATE function won't get the job done!

```
High Ticket Orders =
    CALCULATE(
        [Total Orders],
        'Product Lookup'[ProductPrice] > 714.44 ))
```

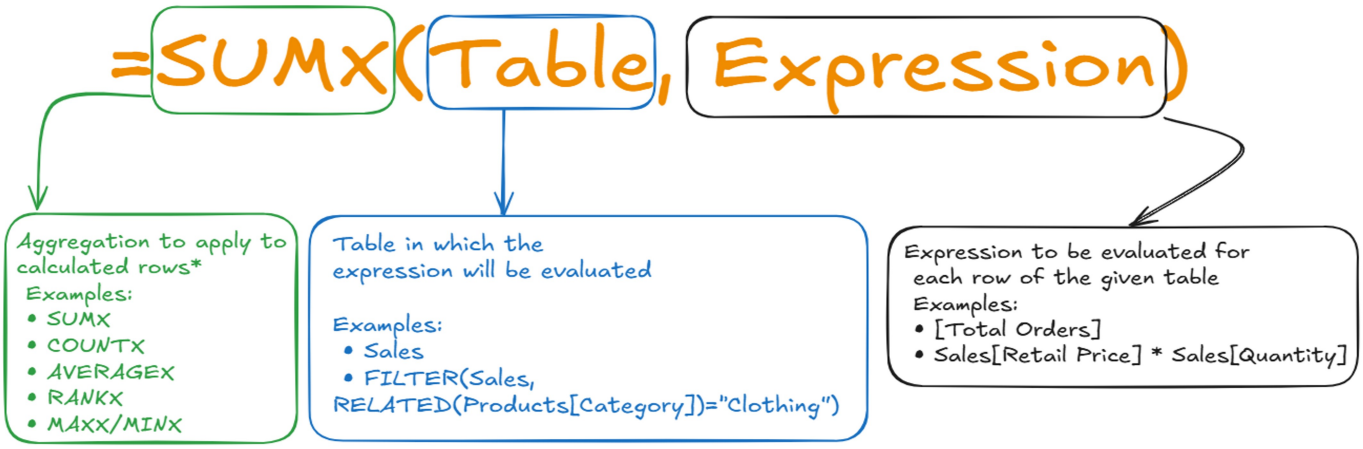
Instead of calling the hard coded value, we must use some filter function to calculate the values row basis.

```
High Ticket Orders =
    CALCULATE(
        [Total Orders],
        FILTER(
            'Product Lookup',
            'Product Lookup'[ProductPrice] > [Overall Average Price]))
```

| CategoryName | Weekend Orders | Total Orders | All Orders | % of All Orders | Average Retail Price | Overall Average Price | High Ticket Orders |
|---|---|---|---|---|---|---|---|
| | | 1 | 25,165 | 0.00% | | $714.44 | |
| Accessories | 4,913 | 16,983 | 25,165 | 67.49% | $34.26 | $714.44 | |
| Bikes | 3,995 | 13,929 | 25,165 | 55.35% | $1,541.38 | $714.44 | 11312 |
| Clothing | 1,962 | 6,976 | 25,165 | 27.72% | $50.68 | $714.44 | |
| Components | | | 25,165 | | $432.19 | $714.44 | |
| **Total** | **7,214** | **25,165** | **25,165** | **100.00%** | **$714.44** | **$714.44** | **11312** |

## ITERATOR FUNCTIONS

Iterator (or "X") functions allow you to loop through the same expression on each row of a table, then apply some sort of aggregation to the results (SUM, MAX, etc.)




PRO TIP:
- Imagine that iterator functions add a temporary new column to a table, calculate a value in each row based on the given expression, then aggregate the values within that temporary column (similar to SUMPRODUCT in Excel).

| Product Price | Quantity | Overall Price |
|---|---|---|
| 100 | 10 | 1,000 |
| 200 | 5 | 1,000 |
| 400 | 5 | 2,000 |
| 500 | 2 | 1,000 |
| 1000 | 1 | 1,000 |

= 6,000

Total Cost = (100 * 10) + (200* 5) + (400 * 5) + (500*2) + (1000 * 1) = 6,000

```
Total Revenue =
    SUMX(
        'Sales Data',
        'Sales Data'[OrderQuantity] *
        RELATED(
            'Product Lookup'[ProductPrice]))
```

$24.91M
Revenue

# ASSIGNMENT: ITERATORS

1. Create a new measure named Total Cost that multiplies the order quantities in the Sales Data table by the product cost in the Product Lookup table, then calculates the sum

$14.46M

Total Cost

```
Total Cost = SUMX('Sales Data',
                  'Sales Data'[OrderQuantity] *
                  RELATED('Product Lookup'[ProductCost]))
```

2. Create a new measure named Total Profit (revenue minus cost)

$10.46M

Total Profit

```
Total Profit = [Revenue] - [Total Cost]
```

3. Create a matrix to show Total Profit (values) by Year (rows). How much profit has AdventureWorks earned so far in 2022?

| Start of Year | Revenue | Total Cost | Total Profit | Total Orders |
|---|---|---|---|---|
| 01-01-2020 | $64,04,933.58 | $38,03,331.25 | $26,01,602.33 | 2630 |
| 01-01-2021 | $93,24,203.79 | $53,57,119.66 | $39,67,084.13 | 10695 |
| 01-01-2022 | $91,85,449.45 | $52,96,420.47 | $38,89,028.97 | 11839 |
| **Total** | **$2,49,14,586.82** | **$1,44,56,871.39** | **$1,04,57,715.43** | **25164** |

These are all unnecessary Calculated Column

```
1 Profit = 'Sales Data'[Revenue] - 'Sales Data'[Expenditure]
```

| ProductKey | CustomerKey | TerritoryKey | OrderLineItem | OrderQuantity | Index | Quantity Type | Product Cost | Product Price | Revenue | Expenditure | Profit |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 371 | 21193 | 9 | 1 | 1 | 1176 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 21173 | 9 | 1 | 1 | 1184 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 21197 | 9 | 1 | 1 | 1287 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 21180 | 9 | 1 | 1 | 1348 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 21202 | 9 | 1 | 1 | 1407 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 21258 | 9 | 1 | 1 | 1466 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 21225 | 9 | 1 | 1 | 1489 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 21418 | 9 | 1 | 1 | 1503 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 21256 | 9 | 1 | 1 | 1575 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 21384 | 9 | 1 | 1 | 1589 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 21219 | 9 | 1 | 1 | 1617 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 21568 | 9 | 1 | 1 | 1714 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 21895 | 9 | 1 | 1 | 1722 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 21886 | 9 | 1 | 1 | 1765 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 21878 | 9 | 1 | 1 | 1783 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 21874 | 9 | 1 | 1 | 1812 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 21873 | 9 | 1 | 1 | 1851 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 21949 | 9 | 1 | 1 | 1855 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 21894 | 9 | 1 | 1 | 1861 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 22172 | 9 | 1 | 1 | 1908 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 21985 | 9 | 1 | 1 | 1924 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 22197 | 9 | 1 | 1 | 1940 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 22173 | 9 | 1 | 1 | 1947 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 21986 | 9 | 1 | 1 | 1951 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 21979 | 9 | 1 | 1 | 1954 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 22033 | 9 | 1 | 1 | 1975 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 22200 | 9 | 1 | 1 | 1976 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 21996 | 9 | 1 | 1 | 1993 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |
| 371 | 21969 | 9 | 1 | 1 | 2095 | Single Item | $1,320.68 | $2,181.56 | $2,181.56 | $1,320.68 | $860.88 |

1st 2nd 3rd 4th 5th 6th 7th 8th 9th 10th 11th 12th 13th 14th 15th 16th 17th