ADVANCED DAX

For

BUISNESS INTELLIGENCE

LinkedIn

Portfolio

DAX Measures

1)

```
Customer Sales =
SUMX(
    'Sales by Store',
    'Sales by Store'[quantity sold] * 'Sales by Store'[unit price]
)
```

store id	Customer Sales
3	$1,389,862.79
5	$1,379,988.03
8	$1,390,892.62
Total	**$4,160,743.44**

2)

```
Cost =
SUMX(
    'Sales by Store',
    'Sales by Store'[quantity sold] *
    RELATED('Product Lookup'[current cost]
    )
)
```

store id	Customer Sales	Cost
3	$1,389,862.79	360,679.29
5	$1,379,988.03	359,192.90
8	$1,390,892.62	360,084.29
Total	**$4,160,743.44**	**1,079,956.48**

3)

```
Profit = [Customer Sales] - [Cost]
```

store id	Customer Sales	Cost	Profit
3	$1,389,862.79	360,679.29	$1,029,183.50
5	$1,379,988.03	359,192.90	$1,020,795.13
8	$1,390,892.62	360,084.29	$1,030,808.33
Total	**$4,160,743.44**	**1,079,956.48**	**$3,080,786.96**

Email: Imamuddinriad@gmail.com

4)

```
Sales for Selected Store =
VAR Store =
SELECTEDVALUE('Store Lookup'[store id]
)

RETURN
CALCULATE(
    SUM(
        'Sales by Store'[quantity sold])*1.05,
        FILTER(
            'Sales by Store',
            'Sales by Store'[store id] = Store
        )
)
```

store id	Sales for Selected Store
3	$445,083.45
5	$450,815.4
8	$445,425.75
Total	

5)

```
Store 3 Sales of Whole Bean/Teas (SUMX) =
SUMX(
    FILTER(
        FILTER(
            'Sales by Store',
            'Sales by Store'[store id] = 3
        ),
        RELATED(
            'Product Lookup'[product group]) = "Whole Bean/Teas"
    ),
    'Sales by Store'[quantity sold] * 'Sales by Store'[unit price]
)
```

store id	Store 3 Sales of Whole Bean/Teas (SUMX)
2	
3	$89,018.04
5	
8	
Total	**$89,018.04**

6)

```
Customer Sales (Last Year) =
VAR LastYearSales =
CALCULATE(
    [Customer Sales],
    DATEADD(
        'Calendar'[Transaction Date],
        -1,
        YEAR
    )
)

VAR Nosale =
IF(
    ISBLANK(
        LastYearSales),
        "No Sales",
        LastYearSales
)

RETURN
NoSale
```

Year ID	Customer Sales	Customer Sales (Last Year - IF-ISBLANK)
2017	$1,678,074.11	No Sales
2018	$1,916,544.75	1,678,074.11
2019	$658,086.02	558,570.72
Total	**$4,252,704.88**	**2,236,644.83**

```
6A)
Customer Sales (Last Year-COALESCE) =
VAR LastYearSales =
CALCULATE(
    [Customer Sales],
    DATEADD(
        'Calendar'[Transaction Date],
        -1,
        YEAR
    )
)

VAR Nosale =
COALESCE(
    LastYearSales,
    "No Sales"
)

RETURN
Nosale
```

Year ID	Customer Sales	Customer Sales (Last Year - IF-ISBLANK)	Customer Sales (Last Year-COALESCE)
2017	$1,678,074.11	No Sales	No Sales
2018	$1,916,544.75	1,678,074.11	1,678,074.11
2019	$658,086.02	558,570.72	558,570.72
Total	**$4,252,704.88**	**2,236,644.83**	**2,236,644.83**

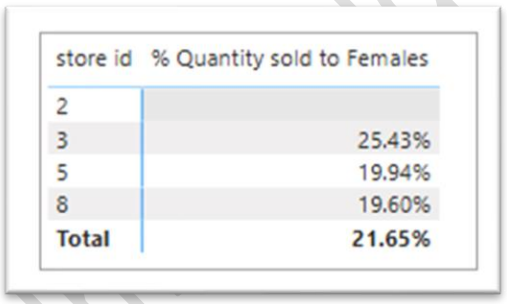
7)

```
Order by Females =
CALCULATE(
    SUM('Sales by Store'[quantity sold]),
    FILTER(
        Customer,
        Customer[gender] = "F"
    )
)
```

store id	Order by Females
2	
F	
M	
Not Specified	
3	**107811**
F	107811
M	
Not Specified	
5	**85600**
F	85600
M	
Not Specified	
8	**83162**
F	83162
M	
Not Specified	
Total	**276573**

8)

```
% Quantity sold to Females =
VAR TotalQuantitySold =
SUM(
    'Sales by Store'[quantity sold]
)

VAR SoldToFemale =
CALCULATE(
    SUM('Sales by Store'[quantity sold]),
    FILTER(
        Customer,
        Customer[gender] = "F"
    )
)

VAR ratio =
DIVIDE(
    SoldToFemale,
    TotalQuantitySold,
    "-"
)

RETURN
ratio
```



store id	% Quantity sold to Females
2	
3	25.43%
5	19.94%
8	19.60%
Total	**21.65%**

SCALAR FUNCTION.

9)

```
Total Customer =
DISTINCTCOUNT(
    Customer[customer_id]
)
```

10)

```
Total Employee =
COUNTROWS(
    Employee
)
```

Year ID	Total Customer	Total Employee
2017	**2251**	**55**
2	2251	55
3	2251	55
5	2251	55
8	2251	55
2018	**2251**	**55**
2	2251	55
3	2251	55
5	2251	55
8	2251	55
2019	**2251**	**55**
2	2251	55
3	2251	55
5	2251	55
8	2251	55
Total	**2251**	**55**

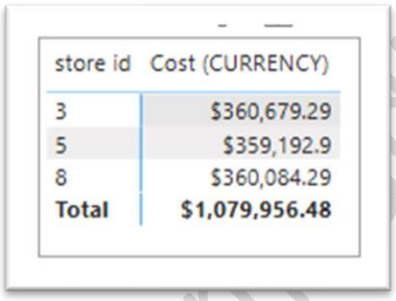
10)

```
Cost =
CURRENCY(
    SUMX(
        'Sales by Store',
        'Sales by Store'[quantity sold] *
        RELATED('Product Lookup'[current cost]
        )
)
)
```

11) Cost (CURRENCY) =

```
ROUND(
    CURRENCY(
            [Cost]
        ),
        2
)
```

comments: 10 should be work, but unfortunately it doesnot change the format into currency directly. We have to do it as shown as 11



store id	Cost (CURRENCY)
3	$360,679.29
5	$359,192.9
8	$360,084.29
Total	**$1,079,956.48**

12)

```
Sum of Quantity sold(without CALCULATE) =
SUM(
    'Food Inventory'[quantity sold]
)
```

12A)

```
Sum of Quantity Sold (CALCULATE) =

CALCULATE(
    SUM(
        'Food Inventory'[quantity sold]
    )
)
```

store id	Sum of Quantity sold(without CALCULATE)	Sum of Quantity Sold (CALCULATE)
2		
3	44,656	44,656
5	48,002	48,002
8	45,705	45,705
Total	**138,363**	**138,363**

11 and 12 gives exact same result when calculates as MEASURE. Example of how Measure always evaluates filter context unlike calculated column

CALCULATE Modifier function

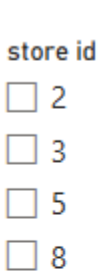
13)REMOVEFILTERS–same as ALL() function

```
Total Profit (REMOVEFILTERS) =
CALCULATE(
    [Profit],
    REMOVEFILTERS(
        'Sales by Store'
    )
)
```

Year ID	Total Profit (REMOVEFILTERS)
2017	**$3,148,872.56**
2	$3,148,872.56
3	$3,148,872.56
5	$3,148,872.56
8	$3,148,872.56
2018	**$3,148,872.56**
2	$3,148,872.56
3	$3,148,872.56
5	$3,148,872.56
8	$3,148,872.56
2019	**$3,148,872.56**
2	$3,148,872.56
3	$3,148,872.56
5	$3,148,872.56
8	$3,148,872.56
Total	**$3,148,872.56**

14)

```
Store 5 Profit =
CALCULATE(
    [Profit],
    'Store Lookup'[store id] = 5
)
```

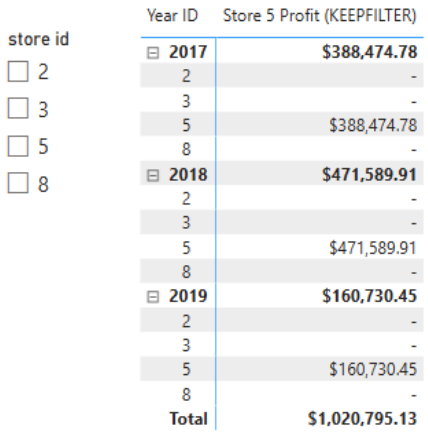


Year ID	Store 5 Profit
2017	**388,474.78**
2	388,474.78
3	388,474.78
5	388,474.78
8	388,474.78
2018	**471,589.91**
2	471,589.91
3	471,589.91
5	471,589.91
8	471,589.91
2019	**160,730.45**
2	160,730.45
3	160,730.45
5	160,730.45
8	160,730.45
Total	**1,020,795.13**

14 will always and only show total sales even if another store from slicer is selected.

14a) KEEPFILTERS

```
Store 5 Profit (KEEPFILTER) =

CALCULATE(
    [Profit],
    KEEPFILTERS(
        'Store Lookup'[store id] = 5
    )
)
```



Year ID	Store 5 Profit (KEEPFILTER)
2017	**$388,474.78**
2	-
3	-
5	$388,474.78
8	-
2018	**$471,589.91**
2	-
3	-
5	$471,589.91
8	-
2019	**$160,730.45**
2	-
3	-
5	$160,730.45
8	-
Total	**$1,020,795.13**

Year ID	Store 5 Profit (KEEPFILTER)
2017	-
3	-
2018	-
3	-
2019	-
3	-
Total	-

14 will always and only show total sales even if another store from slicer is selected.
On the other hand 14a will show store 5 sales if no filter from store id is applied. And if any store except 5 is selected, it will gives blank result.

14b)

```
Store 3 Profit (KEEPFILTERS) =
CALCULATE(
    [Profit],
    KEEPFILTERS(
        'Store Lookup'[store id] = 3
    )
)
```

Year ID	Profit	Store 3 Profit (KEEPFILTERS)
2017	**$1,174,437.09**	**$390,890.14**
3	$390,890.14	$390,890.14
5	$388,474.78	
8	$395,072.18	
2018	**$1,419,047.46**	**$474,697.02**
3	$474,697.02	$474,697.02
5	$471,589.91	
8	$472,760.52	
2019	**$487,302.41**	**$163,596.33**
3	$163,596.33	$163,596.33
5	$160,730.45	
8	$162,975.63	
Total	**$3,080,786.96**	**$1,029,183.50**

store id
- ☐ 2
- ☐ 3
- ☐ 5
- ☐ 8

Year ID	Profit	Store 3 Profit (KEEPFILTERS)
2017	**$388,474.78**	
5	$388,474.78	
2018	**$471,589.91**	
5	$471,589.91	
2019	**$160,730.45**	
5	$160,730.45	
Total	**$1,020,795.13**	

store id
- ☐ 2
- ☐ 3
- ■ 5
- ☐ 8

14c)

```
Store 8 Profit (KEEPFILTERS) =

CALCULATE(
    [Profit],
    KEEPFILTERS(
        'Store Lookup'[store id] = 8
    )
)
```

Year ID	Profit	Store 8 Profit (KEEPFILTERS)
2017	**$1,174,437.09**	**$395,072.18**
3	$390,890.14	
5	$388,474.78	
8	$395,072.18	$395,072.18
2018	**$1,419,047.46**	**$472,760.52**
3	$474,697.02	
5	$471,589.91	
8	$472,760.52	$472,760.52
2019	**$487,302.41**	**$162,975.63**
3	$163,596.33	
5	$160,730.45	
8	$162,975.63	$162,975.63
Total	**$3,080,786.96**	**$1,030,808.33**

store id

- [ ] 2
- [ ] 3
- [ ] 5
- [ ] 8

Year ID	Store 3 Profit (KEEPFILTERS)	Store 5 Profit (KEEPFILTER)	Store 8 Profit (KEEPFILTERS)
2018	**$474,697.02**	**$471,589.91**	**$472,760.52**
2		-	
3	$474,697.02	-	
8		-	$472,760.52
5		$471,589.91	
2017	**$390,890.14**	**$388,474.78**	**$395,072.18**
2		-	
3	$390,890.14	-	
8		-	$395,072.18
5		$388,474.78	
2019	**$163,596.33**	**$160,730.45**	**$162,975.63**
2		-	
3	$163,596.33	-	
8		-	$162,975.63
5		$160,730.45	
Total	**$1,029,183.50**	**$1,020,795.13**	**$1,030,808.33**

15)
% of Store Profit =

```
var TotalProfit =
CALCULATE(
    [Profit],
    REMOVEFILTERS(
       'Store Lookup'[store id]
    )
)


RETURN
DIVIDE(
    [Profit],
   TotalProfit
)
```

Year ID	% of Store Profit
⊟ 2017	100.00%
3	33.28%
5	33.08%
8	33.64%
⊟ 2018	100.00%
3	33.45%
5	33.23%
8	33.32%
⊟ 2019	100.00%
3	33.57%
5	32.98%
8	33.44%
Total	100.00%

16)

```
CUMULATIVE Total (SALES) =
CALCULATE(
    [Customer Sales],
    ALL(
        'Calendar'
    ),
    'Calendar'[Transaction Date] <= MAX('Calendar'[Transaction Date])
)
```

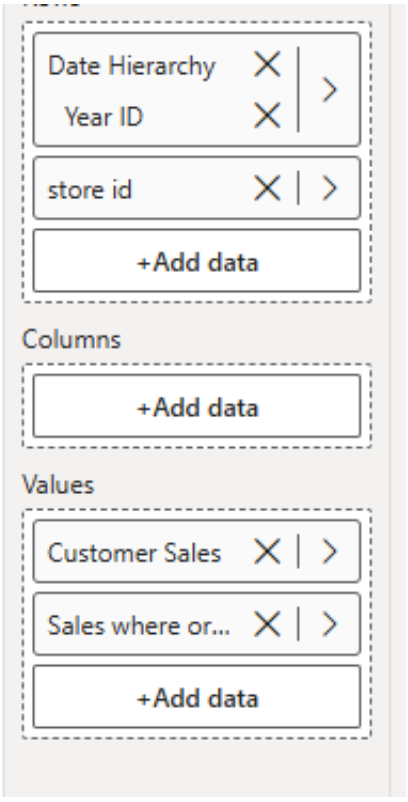
Year ID	Customer Sales	CUMULATIVE Total (SALES)
2017	**$1,678,074.11**	**$1,678,074.11**
1/1/2017	$2,508.20	$2,508.20
1/2/2017	$2,403.35	$4,911.55
1/3/2017	$2,565.00	$7,476.55
1/4/2017	$2,220.10	$9,696.65
1/5/2017	$2,418.85	$12,115.50
1/6/2017	$2,273.85	$14,389.35
1/7/2017	$2,787.00	$17,176.35
1/8/2017	$2,638.53	$19,814.88
1/9/2017	$2,676.61	$22,491.49
1/10/2017	$2,685.65	$25,177.14
1/11/2017	$2,555.75	$27,732.89
1/12/2017	$2,327.70	$30,060.59
1/13/2017	$3,033.60	$33,094.19
Total	**$4,252,704.88**	**$4,252,704.88**

12/26/2017	$5,733.48	$1,652,667.02
12/27/2017	$5,932.53	$1,658,599.55
12/28/2017	$4,773.40	$1,663,372.95
12/29/2017	$4,479.15	$1,667,852.10
12/30/2017	$5,184.03	$1,673,036.13
12/31/2017	$5,037.98	$1,678,074.11
2018	**$1,916,544.75**	**$3,594,618.86**
1/1/2018	$4,360.80	$1,682,434.91
1/2/2018	$4,071.05	$1,686,505.96
1/3/2018	$4,203.75	$1,690,709.71

TABLE and FILTER Function

```
17)
Sales where order quantity more than 3 =
CALCULATE(
    [Customer Sales],
    FILTER(
        'Sales by Store',
        'Sales by Store'[quantity sold] > 3
    )
)
```

Year ID	Customer Sales	Sales where order quantity more than 3
2017	**$1,678,074.11**	**8,670.60**
3	$558,848.52	
5	$554,987.17	1,110.60
8	$564,238.42	7,560.00
2018	**$1,916,544.75**	**8,953.00**
3	$641,008.00	
5	$637,522.57	1,753.00
8	$638,014.18	7,200.00
2019	**$658,086.02**	**2,659.00**
3	$220,864.58	
5	$217,286.77	499.00
8	$219,934.67	2,160.00
Total	**$4,252,704.88**	**20,282.60**



18)

```
Retail Price (SELECTEDVALUE) =
SELECTEDVALUE(
    'Product Lookup'[current_retail_price],
    "-"
)
```

product	Retail Price (SELECTEDVALUE)
Almond Croissant	3.75
Brazilian - Organic	18.00
Brazilian Lg	3.50
Brazilian Rg	3.00
Brazilian Sm	2.20
Cappuccino	3.75
Cappuccino Lg	4.25
Carmel syrup	0.80
Chili Mayan	13.33
Chocolate Chip Biscotti	3.50
Chocolate Croissant	3.75
Chocolate syrup	0.76
Civet Cat	45.00
Columbian Medium Roast	15.00
Columbian Medium Roast Lg	3.00
Total	**-**

19)

```
Quantity Sold (SELECTEDVALUE) =
DIVIDE(
    [Customer Sales],
    SELECTEDVALUE(
        'Product Lookup'[current_retail_price]
    )
)
```

product	Quantity Sold (SELECTEDVALUE)
Almond Croissant	11,757
Brazilian - Organic	1,289
Brazilian Lg	26,001
Brazilian Rg	26,852
Brazilian Sm	26,671
Cappuccino	26,355
Cappuccino Lg	25,836
Carmel syrup	15,458
Chili Mayan	872
Chocolate Chip Biscotti	11,596
Chocolate Croissant	19,240
Chocolate syrup	16,896
Civet Cat	1,479
Columbian Medium Roast	908
Columbian Medium Roast Lg	25,970
Columbian Medium Roast Rg	27,514
Total	

product	Retail Price (SELECTEDVALUE)	Quantity Sold (SELECTEDVALUE)	Customer Sales
Almond Croissant	3.75	11,757	$44,089.73
Brazilian - Organic	18.00	1,289	$23,202.00
Brazilian Lg	3.50	26,001	$91,003.50
Brazilian Rg	3.00	26,852	$80,556.00
Brazilian Sm	2.20	26,671	$58,676.20
Cappuccino	3.75	26,355	$98,831.25
Cappuccino Lg	4.25	25,836	$109,803.00
Carmel syrup	0.80	15,458	$12,366.40
Chili Mayan	13.33	872	$11,623.76
Chocolate Chip Biscotti	3.50	11,596	$40,587.38
Chocolate Croissant	3.75	19,240	$72,150.27
Chocolate syrup	0.76	16,896	$12,840.80
Total	**-**		**$4,252,704.88**

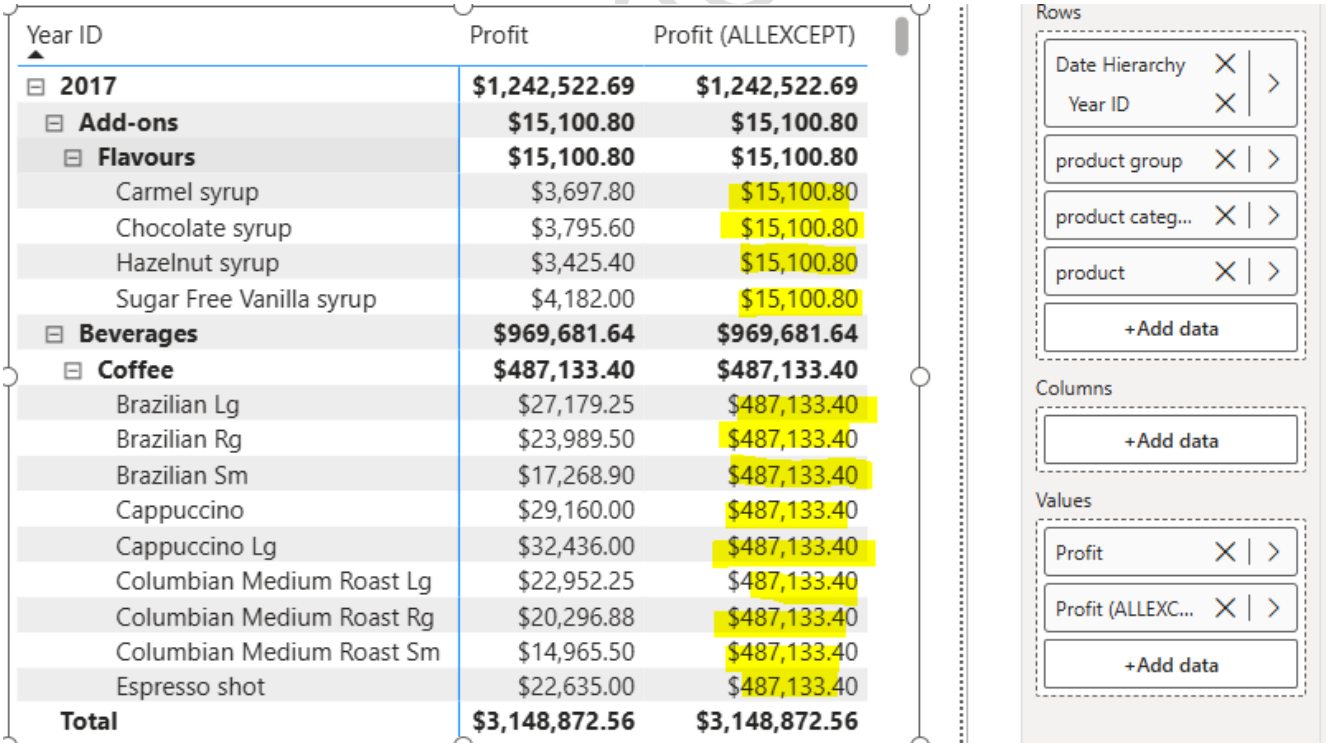
The measure was calculated to better understand the usage of the SELECTEDVALUE function. Assuming we don't have any direct information about the QuantitySold, but we do have the TotalSales and RetailPricePerProduct data, we can calculate the QuantitySold by dividing the IndividualSales by the IndividualRetailPrice. In this scenario, the SELECTEDVALUE function plays a crucial role in

retrieving the individual RetailPrice for each product, helping us correctly calculate the quantity sold.

20)

```
Profit (ALLEXCEPT) =
CALCULATE(
    [Profit],
    ALLEXCEPT(
        'Sales by Store',
        'Product Lookup'[product group],
        'Product Lookup'[product category],
        'Calendar'[Year ID]
    )
)
```

Any filters will be excluded, except for those applied to the "Product Group", "Product Category", and "Year ID" columns. This ensures that only these specific filters are considered in the calculation, while all others are ignored.



Year ID	Profit	Profit (ALLEXCEPT)
2017	**$1,242,522.69**	**$1,242,522.69**
Add-ons	**$15,100.80**	**$15,100.80**
Flavours	**$15,100.80**	**$15,100.80**
Carmel syrup	$3,697.80	$15,100.80
Chocolate syrup	$3,795.60	$15,100.80
Hazelnut syrup	$3,425.40	$15,100.80
Sugar Free Vanilla syrup	$4,182.00	$15,100.80
Beverages	**$969,681.64**	**$969,681.64**
Coffee	**$487,133.40**	**$487,133.40**
Brazilian Lg	$27,179.25	$487,133.40
Brazilian Rg	$23,989.50	$487,133.40
Brazilian Sm	$17,268.90	$487,133.40
Cappuccino	$29,160.00	$487,133.40
Cappuccino Lg	$32,436.00	$487,133.40
Columbian Medium Roast Lg	$22,952.25	$487,133.40
Columbian Medium Roast Rg	$20,296.88	$487,133.40
Columbian Medium Roast Sm	$14,965.50	$487,133.40
Espresso shot	$22,635.00	$487,133.40
Total	**$3,148,872.56**	**$3,148,872.56**

```
21)
Customer Sales (ALLEXCEPT) =
CALCULATE(
    [Customer Sales],
    ALLEXCEPT(
        'Sales by Store',
        'Store Lookup'[store id],
        'Product Lookup'[product group],
        'Calendar'[Transaction Date],
        Customer[customer_first-name]
    )
)
```

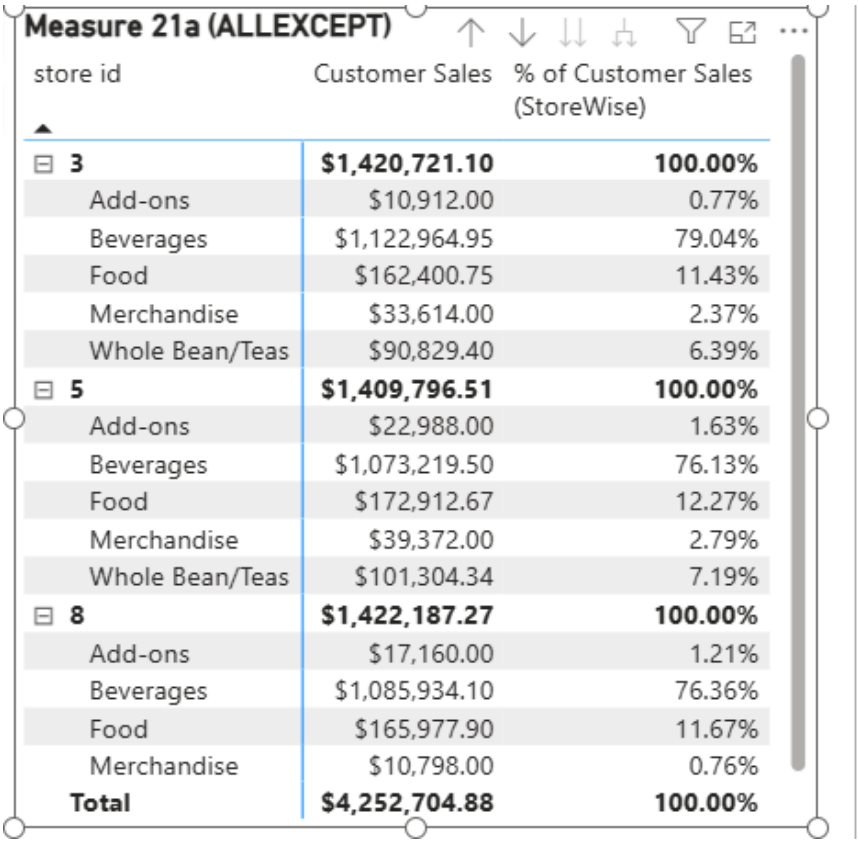
Measure 21(ALLEXCEPT)

Year ID	Customer Sales	Customer Sales (ALLEXCEPT)
2017	**$1,678,074.11**	**$1,678,074.11**
3	**$558,848.52**	**$558,848.52**
Add-ons	**$4,297.60**	**$4,297.60**
Adria Joyner	**$34.40**	**$34.40**
Carmel syrup	$34.40	$34.40
Chocolate syrup		$34.40
Hazelnut syrup		$34.40
Sugar Free Vanilla syrup		$34.40
Adrian Carson	**$12.00**	**$12.00**
Carmel syrup		$12.00
Chocolate syrup		$12.00
Hazelnut syrup		$12.00
Sugar Free Vanilla syrup	$12.00	$12.00
Alec Emerson	**$16.00**	**$16.00**
Carmel syrup		$16.00
Chocolate syrup	$16.00	$16.00
Hazelnut syrup		$16.00
Sugar Free Vanilla syrup		$16.00
Alvin Howe	**$16.00**	**$16.00**
Carmel syrup		$16.00
Total	**$4,252,704.88**	**$4,252,704.88**

This DAX measure calculates **Customer Sales** while keeping filters only on the **Store ID**, **Product Group**, **Transaction Date**, and **Customer First Name** columns. The **ALLEXCEPT** function ensures that filters on these columns are preserved, while any other filters are ignored. As a result, at lower levels of granularity (e.g., when drilling down to the **Product** level), the values for **Customer Sales** will repeat based on the **Customer First Name** because the filter context for other columns (like **Product**) is cleared. This leads to repeating values when the measure is evaluated at more detailed levels within the hierarchy**.**

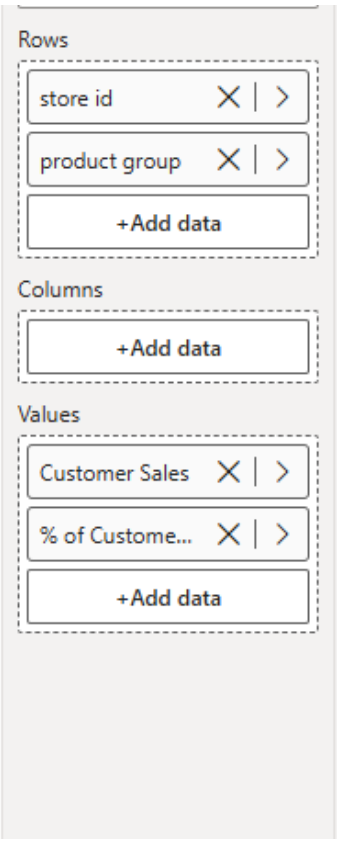
21a)
% of Customer Sales (StoreWise) =

```
VAR AllSales =
CALCULATE(
    [Customer Sales],
    ALLEXCEPT(
        'Sales by Store',
        'Store Lookup'[store id]
    )
)  -- Gives Total sales in a store, excluding other filters.

VAR ratio =
DIVIDE(
    [Customer Sales],
    AllSales
)
RETURN
AllSales
```



Measure 21a (ALLEXCEPT)

store id	Customer Sales	% of Customer Sales (StoreWise)
3	**$1,420,721.10**	**100.00%**
Add-ons	$10,912.00	0.77%
Beverages	$1,122,964.95	79.04%
Food	$162,400.75	11.43%
Merchandise	$33,614.00	2.37%
Whole Bean/Teas	$90,829.40	6.39%
5	**$1,409,796.51**	**100.00%**
Add-ons	$22,988.00	1.63%
Beverages	$1,073,219.50	76.13%
Food	$172,912.67	12.27%
Merchandise	$39,372.00	2.79%
Whole Bean/Teas	$101,304.34	7.19%
8	**$1,422,187.27**	**100.00%**
Add-ons	$17,160.00	1.21%
Beverages	$1,085,934.10	76.36%
Food	$165,977.90	11.67%
Merchandise	$10,798.00	0.76%
Total	**$4,252,704.88**	**100.00%**

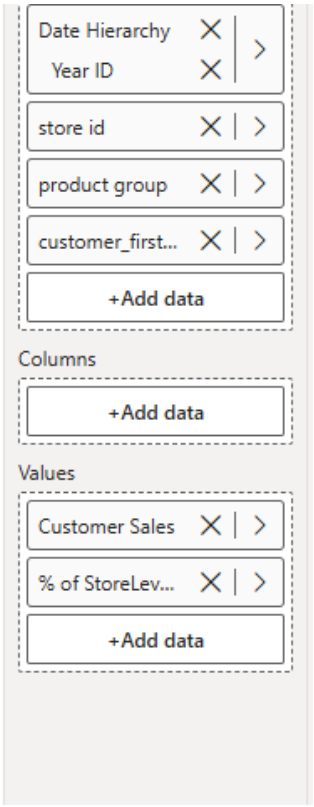


22)
% of StoreLevelSales (ALLEXCEPT) =

```
VAR storelevelsales =
CALCULATE(
    [Customer Sales],
    REMOVEFILTERS(
        Customer
    )
) --Any filter from customer table will be excluded

RETURN
DIVIDE(
    [Customer Sales],
    Storelevelsales
)
```

Measure 22 (ALLEXCEPT)

Year ID	Customer Sales	% of StoreLevelSales (ALLEXCEPT)
2017	**$1,678,074.11**	**100.00%**
3	**$558,848.52**	**100.00%**
Add-ons	**$4,297.60**	**100.00%**
Adria Joyner	$34.40	0.80%
Adrian Carson	$12.00	0.28%
Alec Emerson	$16.00	0.37%
Alvin Howe	$16.00	0.37%
Alvin Mann	$5.60	0.13%
Amanda Workman	$24.80	0.58%
Amela Watts	$6.40	0.15%
Amir Byers	$11.20	0.26%
Angelica Maynard	$11.20	0.26%
Aquila Strong	$17.60	0.41%
Ariana Howell	$14.40	0.34%
Aurelia Bernard	$5.60	0.13%
Avram Cooper	$11.20	0.26%
Baker Turner	$16.00	0.37%
Beatrice Gutierrez	$25.60	0.60%
Total	**$4,252,704.88**	**100.00%**



23)

```
Customer Sales (ALLSELECTED) =
CALCULATE(
    [Customer Sales],
    ALLSELECTED()
)
```

ALLSELECTED ignores filter from row\column levels, but respects filter from outside(slicer)

$1.42M

Customer Sales for store 3

Measure 23 (ALLSELECTED)

Year ID	Customer Sales	Customer Sales (ALLSELECTED)
2017	**$558,848.52**	**$1,420,721.10**
3	**$558,848.52**	**$1,420,721.10**
Add-ons	$4,297.60	$1,420,721.10
Beverages	$442,201.25	$1,420,721.10
Food	$64,135.50	$1,420,721.10
Merchandise	$12,986.00	$1,420,721.10
Whole Bean/Teas	$35,228.17	$1,420,721.10
2018	**$641,008.00**	**$1,420,721.10**
3	**$641,008.00**	**$1,420,721.10**
Add-ons	$4,951.20	$1,420,721.10
Beverages	$506,841.50	$1,420,721.10
Food	$73,122.50	$1,420,721.10
Merchandise	$15,227.00	$1,420,721.10
Whole Bean/Teas	$40,865.80	$1,420,721.10
2019	**$220,864.58**	**$1,420,721.10**
3	**$220,864.58**	**$1,420,721.10**
Add-ons	$1,663.20	$1,420,721.10
Beverages	$173,922.20	$1,420,721.10
Food	$25,142.75	$1,420,721.10
Merchandise	$5,401.00	$1,420,721.10
Whole Bean/Teas	$14,735.43	$1,420,721.10
Total	**$1,420,721.10**	**$1,420,721.10**

Measure 23 (ALLSELECTED)

Year ID	Customer Sales	Customer Sales (ALLSELECTED)
2017	**$554,987.17**	**$1,409,796.51**
5	**$554,987.17**	**$1,409,796.51**
Add-ons	$8,932.00	$1,409,796.51
Beverages	$421,751.85	$1,409,796.51
Food	$68,394.73	$1,409,796.51
Merchandise	$15,540.00	$1,409,796.51
Whole Bean/Teas	$40,368.59	$1,409,796.51
2018	**$637,522.57**	**$1,409,796.51**
5	**$637,522.57**	**$1,409,796.51**
Add-ons	$10,508.00	$1,409,796.51
Beverages	$485,637.35	$1,409,796.51
Food	$78,012.52	$1,409,796.51
Merchandise	$17,828.00	$1,409,796.51
Whole Bean/Teas	$45,536.70	$1,409,796.51
2019	**$217,286.77**	**$1,409,796.51**
5	**$217,286.77**	**$1,409,796.51**
Add-ons	$3,548.00	$1,409,796.51
Beverages	$165,830.30	$1,409,796.51
Food	$26,505.42	$1,409,796.51
Merchandise	$6,004.00	$1,409,796.51
Whole Bean/Teas	$15,399.05	$1,409,796.51
Total	**$1,409,796.51**	**$1,409,796.51**

store id

- ☐ 2
- ☐ 3
- ■ 5
- ☐ 8

$1.41M

Customer Sales for store 5

23a)

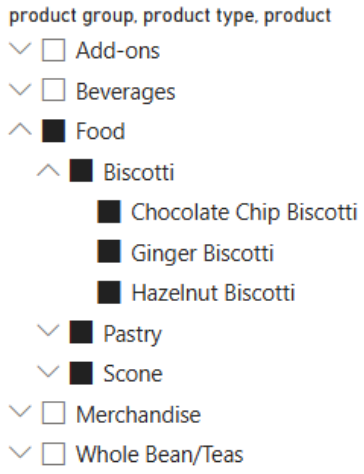
```
Total Backed (Food Inventory) =
SUM(
    'Food Inventory'[quantity start of day]
)
```

23b)

```
Total sold (Food Inventory) =
SUMX(
    'Food Inventory',
    'Food Inventory'[quantity sold]
)
```

24a)

```
% of Total Quantity Backed (ALLSELECTED) =
VAR SelectedProducttotal =
CALCULATE(
    [Total Backed (Food Inventory)],
    ALLSELECTED(
    )
)

VAR ratio =
DIVIDE(
    [Total Backed (Food Inventory)],
    selectedproducttotal
)

RETURN
Ratio
```



Measure 23a,23b,24a,24b (ALLSELECTED)

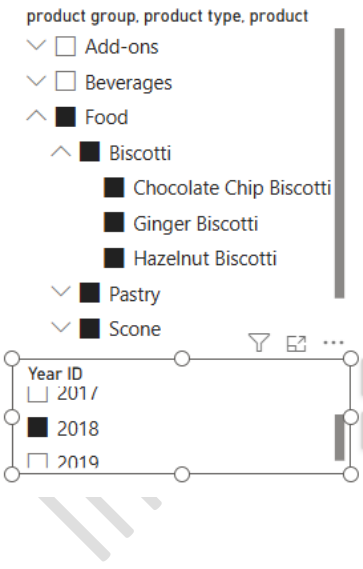
product group	Total Backed (Food Inventory)	Total sold (Food Inventory)	% of Total Quantity Backed (ALLSELECTED)	% of Total Quantity sold (ALLSELECTED)
Food	**648282**	**141433**	**100.00%**	**100.00%**
Biscotti	**136350**	**34921**	**21.03%**	**24.69%**
Chocolate Chip Biscotti	45630	11616	7.04%	8.21%
Ginger Biscotti	45360	11007	7.00%	7.78%
Hazelnut Biscotti	45360	12298	7.00%	8.70%
Pastry	**135486**	**42819**	**20.90%**	**30.28%**
Almond Croissant	44658	11761	6.89%	8.32%
Chocolate Croissant	45630	19217	7.04%	13.59%
Croissant	45198	11841	6.97%	8.37%
Scone	**376446**	**63693**	**58.07%**	**45.03%**
Cranberry Scone	45198	12554	6.97%	8.88%
Ginger Scone	120480	15849	18.58%	11.21%
Jumbo Savory Scone	45630	12544	7.04%	8.87%
Oatmeal Scone	120480	11034	18.58%	7.80%
Scottish Cream Scone	44658	11712	6.89%	8.28%
Total	**648282**	**141433**	**100.00%**	**100.00%**

24b)

```
% of Total Quantity sold (ALLSELECTED) =
VAR QuantitySold_Allselected  =
    CALCULATE(
        [Total Quantity Sold (Food inventory )],
        ALLSELECTED(
           -- 'Product Lookup'
        )
    )


VAR ratio =
DIVIDE(
    [Total Quantity Sold (Food inventory )],
    QuantitySold_Allselected,
    "0"
)

RETURN
ratio
```

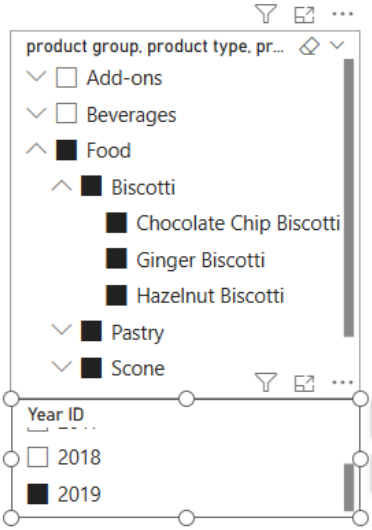
24c)

```
% of AllBacked =

VAR allbacked =
CALCULATE(
    [Total Backed (Food Inventory)],
    REMOVEFILTERS(
        'Food Inventory'
    )
)


VAR ratio =
DIVIDE(
    [Total Backed (Food Inventory)],
    allbacked,
    "-"
)

RETURN
ratio
```



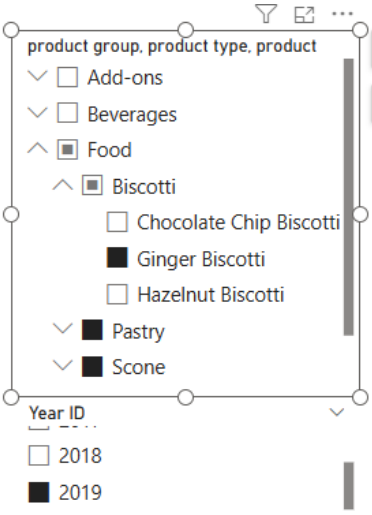
Measure 23a,23b,24a,24b,24c (ALLSELECTED)

product group	Total Backed (Food Inventory)	Total sold (Food Inventory)	% of Total Quantity Backed (ALLSELECTED)	% of Total Quantity sold (ALLSELECTED)	% of AllBacked
Food	**279978**	**63670**	**100.00%**	**100.00%**	**43.19%**
Biscotti	**58662**	**15650**	**20.95%**	**24.58%**	**9.05%**
Chocolate Chip Biscotti	19638	5186	7.01%	8.15%	3.03%
Ginger Biscotti	19512	4952	6.97%	7.78%	3.01%
Hazelnut Biscotti	19512	5512	6.97%	8.66%	3.01%
Pastry	**58482**	**19374**	**20.89%**	**30.43%**	**9.02%**
Almond Croissant	19314	5312	6.90%	8.34%	2.98%
Chocolate Croissant	19638	8706	7.01%	13.67%	3.03%
Croissant	19530	5356	6.98%	8.41%	3.01%
Scone	**162834**	**28646**	**58.16%**	**44.99%**	**25.12%**
Cranberry Scone	19530	5626	6.98%	8.84%	3.01%
Ginger Scone	52176	7199	18.64%	11.31%	8.05%
Jumbo Savory Scone	19638	5650	7.01%	8.87%	3.03%
Oatmeal Scone	52176	4973	18.64%	7.81%	8.05%
Scottish Cream Scone	19314	5198	6.90%	8.16%	2.98%
Total	**279978**	**63670**	**100.00%**	**100.00%**	**43.19%**



Measure 23a,23b,24a,24b,24c (ALLSELECTED)

product group	Total Backed (Food Inventory)	Total sold (Food Inventory)	% of Total Quantity Backed (ALLSELECTED)	% of Total Quantity sold (ALLSELECTED)	% of AllBacked
Food	**91776**	**21820**	**100.00%**	**100.00%**	**14.16%**
Biscotti	**19368**	**5445**	**21.10%**	**24.95%**	**2.99%**
Chocolate Chip Biscotti	6444	1792	7.02%	8.21%	0.99%
Ginger Biscotti	6462	1726	7.04%	7.91%	1.00%
Hazelnut Biscotti	6462	1927	7.04%	8.83%	1.00%
Pastry	**19116**	**6562**	**20.83%**	**30.07%**	**2.95%**
Almond Croissant	6318	1816	6.88%	8.32%	0.97%
Chocolate Croissant	6444	2941	7.02%	13.48%	0.99%
Croissant	6354	1805	6.92%	8.27%	0.98%
Scone	**53292**	**9813**	**58.07%**	**44.97%**	**8.22%**
Cranberry Scone	6354	1896	6.92%	8.69%	0.98%
Ginger Scone	17088	2444	18.62%	11.20%	2.64%
Jumbo Savory Scone	6444	1935	7.02%	8.87%	0.99%
Oatmeal Scone	17088	1734	18.62%	7.95%	2.64%
Scottish Cream Scone	6318	1804	6.88%	8.27%	0.97%
Total	**91776**	**21820**	**100.00%**	**100.00%**	**14.16%**



Measure 23a,23b,24a,24b,24c (ALLSELECTED)

product group	Total Backed (Food Inventory)	Total sold (Food Inventory)	% of Total Quantity Backed (ALLSELECTED)	% of Total Quantity sold (ALLSELECTED)	% of AllBacked
Food	**78870**	**18101**	**100.00%**	**100.00%**	**12.17%**
Biscotti	**6462**	**1726**	**8.19%**	**9.54%**	**1.00%**
Ginger Biscotti	6462	1726	8.19%	9.54%	1.00%
Pastry	**19116**	**6562**	**24.24%**	**36.25%**	**2.95%**
Almond Croissant	6318	1816	8.01%	10.03%	0.97%
Chocolate Croissant	6444	2941	8.17%	16.25%	0.99%
Croissant	6354	1805	8.06%	9.97%	0.98%
Scone	**53292**	**9813**	**67.57%**	**54.21%**	**8.22%**
Cranberry Scone	6354	1896	8.06%	10.47%	0.98%
Ginger Scone	17088	2444	21.67%	13.50%	2.64%
Jumbo Savory Scone	6444	1935	8.17%	10.69%	0.99%
Oatmeal Scone	17088	1734	21.67%	9.58%	2.64%
Scottish Cream Scone	6318	1804	8.01%	9.97%	0.97%
Total	**78870**	**18101**	**100.00%**	**100.00%**	**12.17%**

RELATIONSHIP Function

25)

```
WholeSale Cost =
SUMX(
    'Sales by Store',
    'Sales by Store'[quantity sold] *
    RELATED(
        'Product Lookup'[current wholesale price]
    )
)
```

26)

```
Total Quantity Sold (USERELATIONSHIP) =
CALCULATE(
    SUMX(
        'Food Inventory',
        'Food Inventory'[quantity sold]
    ),
    USERELATIONSHIP(
        'Food Inventory'[baked date],    --Foreign keys
        'Calendar'[Transaction Date]    -- Primary Keys
    )
)
```

Measure 26 (ALLSELECTED)

Year ID	Total sold (Food Inventory)	Total Quantity Sold (USERELATIONSHIP)
2017	$55,943.00	$56,242.00
2018	$63,670.00	$63,668.00
2019	$21,820.00	$21,045.00
Total	**$141,433.00**	**$140,955.00**

```
27)
Number of Employees that generates sales in an individual day
(CROSSFILTER) =
CALCULATE(
    COUNTROWS(
        Employee
    ),
    CROSSFILTER(
        'Sales by Store'[staff id],  -- Many side table
        Employee[staff id],     -- One side table
        Both
    )
)
```

Measure 27 (CROSSFILTER)

Transaction Date	Customer Sales	Number of Employees that generates sales in an individual day (CROSSFILTER)
1/1/2017	$2,508.20	7
1/2/2017	$2,403.35	7
1/3/2017	$2,565.00	7
1/4/2017	$2,220.10	7
1/5/2017	$2,418.85	7
1/6/2017	$2,273.85	7
1/7/2017	$2,787.00	11
1/8/2017	$2,638.53	15
1/9/2017	$2,676.61	16
1/10/2017	$2,685.65	15
1/11/2017	$2,555.75	15
1/12/2017	$2,327.70	15
1/13/2017	$3,033.60	16
Total	**$4,252,704.88**	**25**

Measure 27 (CROSSFILTER)

Transaction Date	Customer Sales	Number of Employees that generates sales in an individual day (CROSSFILTER)
1/1/2017	**$2,508.20**	**7**
3	$868.40	2
5	$788.35	3
8	$851.45	3
1/2/2017	**$2,403.35**	**7**
3	$925.50	2
5	$649.05	3
8	$828.80	3
1/3/2017	**$2,565.00**	**7**
3	$902.75	2
5	$756.00	3
8	$906.25	3
1/4/2017	**$2,220.10**	**7**
Total	**$4,252,704.88**	**25**

```
28)
Number of Customer who purchased (CROSSFILTER) =
CALCULATE(
    COUNTROWS(
        Customer
    ),
    CROSSFILTER(
        'Sales by Store'[customer id],   --Many sided table
        'Customer'[customer_id],   -- One sided Table
        Both
    )
)
```

Measure 28 (CROSSFILTER)

Transaction Date	Customer Sales	Number of Customer who purchased(CROSSFILTER)
1/1/2017	**$2,508.20**	**423**
3	$868.40	155
5	$788.35	134
8	$851.45	134
1/2/2017	**$2,403.35**	**425**
3	$925.50	163
5	$649.05	121
8	$828.80	141
1/3/2017	**$2,565.00**	**449**
3	$902.75	163
5	$756.00	124
8	$906.25	162
1/4/2017	**$2,220.10**	**392**
3	$808.25	149
Total	**$4,252,704.88**	**2250**

```
28a)
AVG OrderValue =
DIVIDE (
    [Customer Sales],
    [Number of Customer who purchased (CROSSFILTER)]
)
```

Measure 28 & 28a (CROSSFILTER)

Transaction Date	Customer Sales	Number of Customer who purchased(CROSSFILTER)	AVG OrderValue
1/1/2017	$2,508.20	423	$5.93
1/2/2017	$2,403.35	425	$5.65
1/3/2017	$2,565.00	449	$5.71
1/4/2017	$2,220.10	392	$5.66
1/5/2017	$2,418.85	435	$5.56
1/6/2017	$2,273.85	385	$5.91
1/7/2017	$2,787.00	249	$11.19
1/8/2017	$2,638.53	214	$12.33
1/9/2017	$2,676.61	168	$15.93
1/10/2017	$2,685.65	235	$11.43
1/11/2017	$2,555.75	249	$10.26
1/12/2017	$2,327.70	207	$11.24
1/13/2017	$3,033.60	231	$13.13
1/14/2017	$2,682.51	115	$23.33
Total	**$4,252,704.88**	**2250**	**$1,890.09**

29a) **TREATAS**

```
Bean/Teas Goal(TREATAS) =
CALCULATE(
    SUMX(
        'X-Demo UNION',
        'X-Demo UNION'[Bean/Teas Goal]
    ),
    TREATAS(
        VALUES(
            'Calendar'[Year ID]
        ),
        'X-Demo UNION'[Year]
    ),
    TREATAS(
        VALUES(
            'Calendar'[Month Name]
        ),
        'X-Demo UNION'[Month]
    )
)
```

29b)

```
% to Goal(Bean/Teas) =
DIVIDE(
    CALCULATE(
        SUMX(
            'Sales by Store',
            'Sales by Store'[quantity sold]
        ),
        'Product Lookup'[product group] = "Whole Bean/Teas"
    ),
    [Bean/Teas Goal(TREATAS)]
)
```

Measure 29a & 29b (TREATAS)

Year ID	Bean/Teas Goal(TREATAS)	% to Goal(Bean/Teas)
⊟ **2019**	**1885**	**177.14%**
March	963	79.75%
April	922	123.64%
Total	**1885**	**1129.50%**

```
29c)
Beverage Goal(TREATAS) =
CALCULATE(
    SUMX(
        'X-Demo UNION',
        'X-Demo UNION'[Beverage Goal]
    ),
    TREATAS(
        SUMMARIZE(
            'Calendar',
            'Calendar'[Year ID],
            'Calendar'[Month Name]
        ),
        'X-Demo UNION'[Year],
        'X-Demo UNION'[Month]
    )
)

29d)
% to Goal(Beverage) =
DIVIDE(
    CALCULATE(
        SUMX(
            'Sales by Store',
            'Sales by Store'[quantity sold]
        ),
        'Product Lookup'[product group] = "Beverages"
    ),
    [Beverage Goal(TREATAS)]
)
```

Measure 29c & 29d (TREATAS)

Year ID	Beverage Goal(TREATAS)	% to Goal(Beverage)
2019	**90912**	**182.87%**
March	45606	82.79%
April	45306	130.03%
Total	**90912**	**1181.79%**

```
29e)
Food Goal (TREATAS) =
CALCULATE(
    SUMX(
        'X-Demo UNION',
        'X-Demo UNION'[Food Goal]
    ),
    TREATAS(
        VALUES(
            'Calendar'[Year ID]
        ),
        'X-Demo UNION'[Year]
    ),
    TREATAS(
        VALUES(
            'Calendar'[Month Name]
        ),
        'X-Demo UNION'[Month]
    )
)

29f)
% to Goal(Food) =
DIVIDE(
    CALCULATE(
        SUMX(
            'Sales by Store',
            'Sales by Store'[quantity sold]
        ),
        'Product Lookup'[product group] = "Food"
    ),
    [Food Goal (TREATAS)]
)
```

Measure 29e & 29f (TREATAS)

Year ID	Food Goal (TREATAS)	% to Goal(Food)
⊟ 2019	11935	182.96%
March	5978	83.04%
April	5957	129.78%
Total	11935	1185.03%

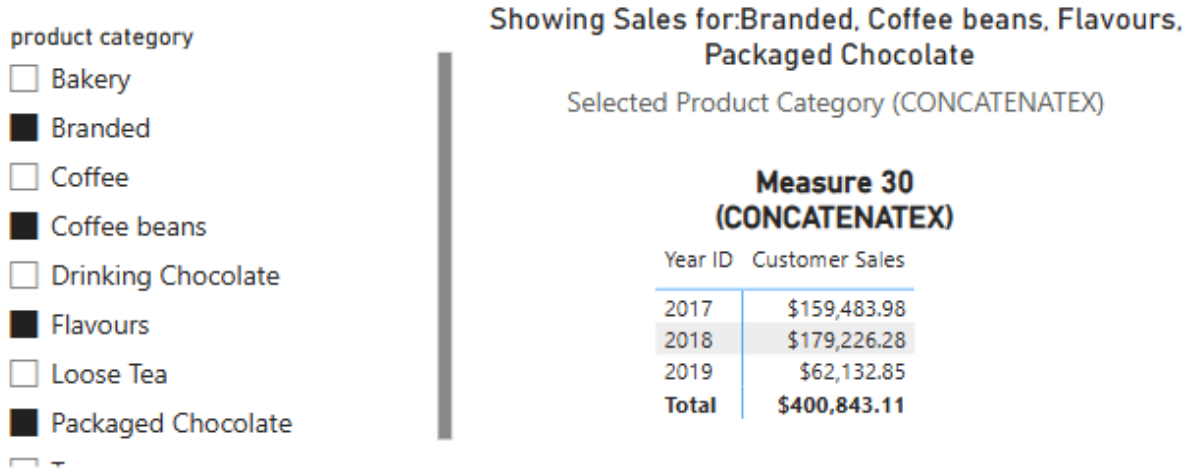
```
29g)
Merchandise Goal(TREATAS) =
CALCULATE(
    SUMX(
        'X-Demo UNION',
        'X-Demo UNION'[Merchandise Goal]
    ),
    TREATAS(
        SUMMARIZE(
            'Calendar',
            'Calendar'[Year ID],
            'Calendar'[Month Name]
        ),
        'X-Demo UNION'[Year],
        'X-Demo UNION'[Month]
    )
)

29h)
% to Goal(Merchandise) =
DIVIDE(
    CALCULATE(
        SUMX(
            'Sales by Store',
            'Sales by Store'[quantity sold]
        ),
        'Product Lookup'[product group] = "Merchandise"
    ),
    [Merchandise Goal(TREATAS)]
)
```

Measure 29g & 29h (TREATAS)

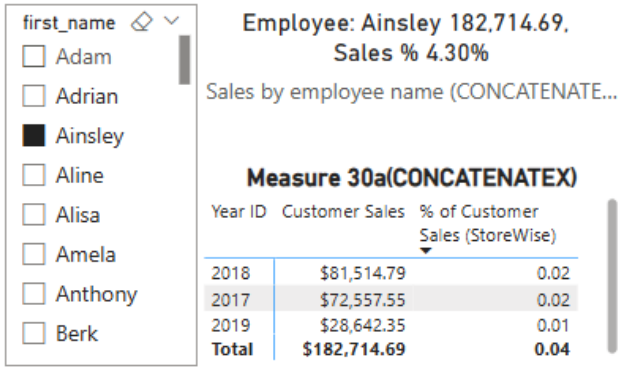
Year ID	Merchandise Goal(TREATAS)	% to Goal(Merchandise)
2019	**423**	**169.50%**
March	218	78.44%
April	205	124.39%
Total	**423**	**1110.40%**

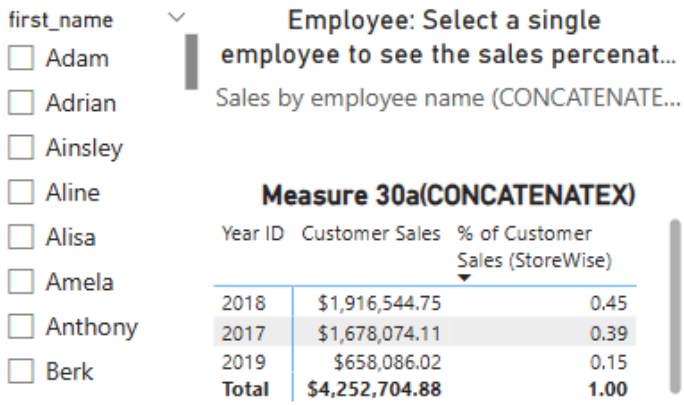
Iterator Fucntion

```
30)
Selected Product Category (CONCATENATEX) =
"Showing Sales for:" &
CONCATENATEX(
    VALUES(
        'Product Lookup'[product category]
    ),
    'Product Lookup'[product category],
    ", ",
    'Product Lookup'[product category],
    ASC
)
```



```
30a)
Sales by employee name (CONCATENATEX) =
"Employee: " &
IF(
    HASONEVALUE(
        Employee[first_name]
    ),
    CONCATENATEX(
       VALUES(
             Employee[first_name]
       ),
       Employee[first_name] & " " & FORMAT([Customer Sales],"#,###.00")
       & " " &  "Sales % " & FORMAT([% of Customer Sales
       (StoreWise)],"Percent"),
       ", ",
      Employee[first_name],
      ASC
       ),
"Select a single employee to see the sales percenatages"
)
```





31)

```
Average Daily Sales (AVERAGEX) =
AVERAGEX(
    'Calendar',
    [Customer Sales]
)
```

Measure 31 (AVERAGEX)

Transaction Date	Average Daily Sales (AVERAGEX)
1/1/2018	$479.00
1/2/2018	$423.00
1/3/2018	$480.50
1/4/2018	$462.50
1/5/2018	$515.75
1/6/2018	$454.00
1/7/2018	$517.75
1/8/2018	$704.50
1/9/2018	$595.37
1/10/2018	$546.00
Total	**$618.78**

31a)

```
Moving Average(AVERAGEX) =
VAR LastTransactionDate = MAX('Calendar'[Transaction Date])

VAR AverageDay = 30

VAR PeriodInVisual =
FILTER(
    ALL(
        'Calendar'
    ),
    AND(
        'Calendar'[Transaction Date] > LastTransactionDate -
         AverageDay,
        'Calendar'[Transaction Date] <= LastTransactionDate
    )
)

VAR Output =
CALCULATE(
    AVERAGEX(
        'Calendar',
        [Customer Sales]
    ),
    PeriodInVisual
)

RETURN
Output
```

Measure 31a (AVERAGEX)

Transaction Date	Customer Sales	Moving Average(AVERAGEX)
1/1/2017	$241.50	241.50
1/2/2017	$309.75	275.63
1/3/2017	$288.75	280.00
1/4/2017	$214.50	263.63
1/5/2017	$259.25	262.75
1/6/2017	$275.75	264.92
1/7/2017	$300.25	269.96
1/8/2017	$365.50	281.91
1/9/2017	$276.04	281.25
1/10/2017	$350.75	288.20
Total	**$501,291.32**	**931.38**

```
31b)
Daily Average Profit =
AVERAGEX(
    'Calendar',
    [Profit]
)
```

Measure 31b (AVERAGEX)

Transaction Date	Daily Average Profit
1/1/2017	148.46
1/2/2017	190.52
1/3/2017	177.02
1/4/2017	129.05
1/5/2017	160.89
1/6/2017	167.41
1/7/2017	182.22
1/8/2017	222.36
1/9/2017	175.89
1/10/2017	215.23
Total	**360.44**

31-C) –TABLEs

■ Creating a table

```
Average Day(GENERATESERIES) =
GENERATESERIES(
    7,
    63,
    7
)
```

Value
7
14
21
28
35
42
49
56
63

■ Creating a measure from the newly created table.

```
Average Days Value =
SELECTEDVALUE(
    'Average Day(GENERATESERIES)'[Value],
    30
)
```

■ Create a measure for Moving average of profit

```
Profit Moving Average (AVERAGEX) =
VAR LastTransactionDate = MAX('Calendar'[Transaction Date])

VAR AverageDay = [Average Days Value]

VAR PeriodInVisual =
FILTER(
    ALL(
        'Calendar'
    ),
    AND(
        'Calendar'[Transaction Date] > LastTransactionDate -
         AverageDay,
        'Calendar'[Transaction Date] <= LastTransactionDate)
)

VAR Output =
CALCULATE(
    AVERAGEX(
        'Calendar',
        [Profit]
    ),
   PeriodInVisual
)

RETURN
Output
```

Value
■ 7
☐ 14
☐ 21
☐ 28
☐ 35
☐ 42
☐ 49
☐ 56
☐ 63

Measure 31c (AVERAGEX)

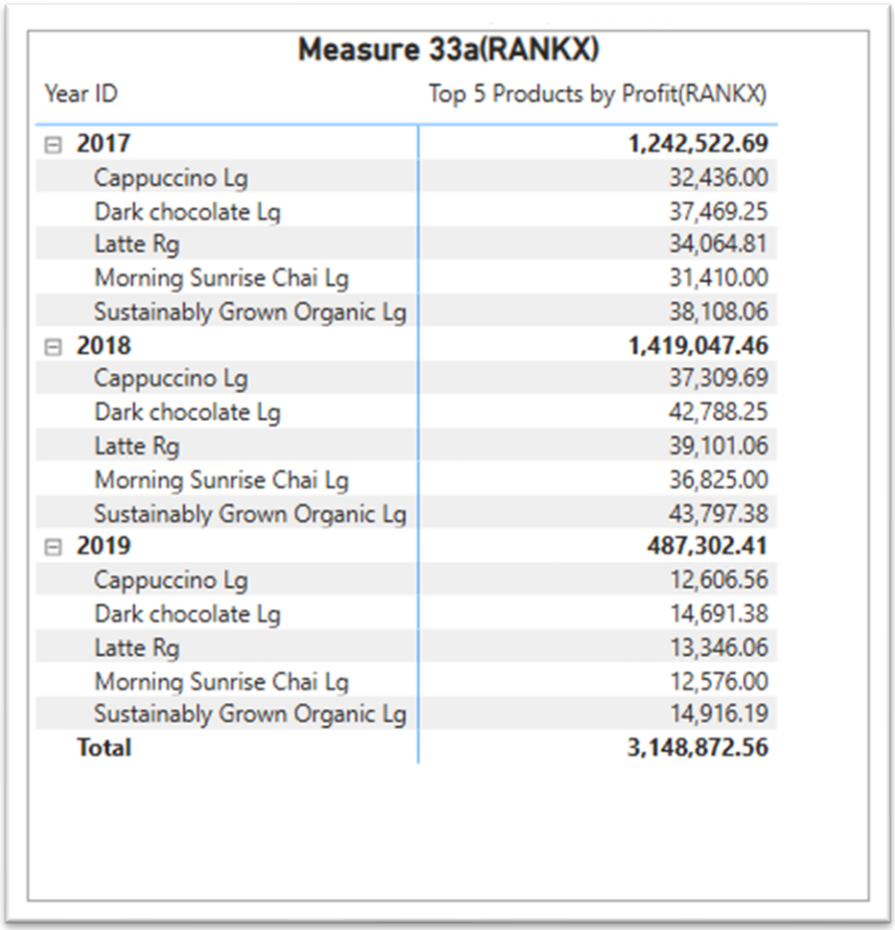
Transaction Date	Profit	Profit Moving Average (AVERAGEX)
1/1/2017	$1,848.48	$1,848.48
1/2/2017	$1,760.72	$1,804.60
1/3/2017	$1,884.21	$1,831.14
1/4/2017	$1,633.25	$1,781.67
1/5/2017	$1,780.59	$1,781.45
1/6/2017	$1,665.98	$1,762.21
1/7/2017	$2,067.37	$1,805.80
1/8/2017	$1,955.68	$1,821.11
1/9/2017	$2,011.82	$1,856.99
1/10/2017	$1,989.38	$1,872.01
1/11/2017	$1,898.29	$1,909.87
1/12/2017	$1,714.80	$1,900.47
1/13/2017	$2,254.11	$1,984.49
1/14/2017	$1,995.54	$1,974.23
1/15/2017	$2,360.10	$2,032.00
1/16/2017	$2,103.85	$2,045.15
1/17/2017	$2,494.78	$2,117.35
1/18/2017	$2,041.67	$2,137.84
1/19/2017	$2,165.47	$2,202.22
1/20/2017	$1,923.00	$2,154.92
Total	**$3,148,872.56**	**$5,848.41**

```
33)
Rank of Customer Sales (RANKX) =
IF(
    HASONEVALUE(
        'Product Lookup'[product category]
    ),
    RANKX(
      ALL(
       'Product Lookup'[product category]
      ),
    [Customer Sales],
    ,
    DESC,
    Dense
),
BLANK()
)
```

Measure 33 (RANKX)

Year ID	Rank of Customer Sales (RANKX)
2017	
Bakery	3
Branded	6
Coffee	1
Coffee beans	5
Drinking Chocolate	4
Flavours	8
Loose Tea	7
Packaged Chocolate	9
Tea	2
2018	
Bakery	3
Branded	6
Coffee	1
Coffee beans	5
Drinking Chocolate	4
Flavours	8
Loose Tea	7
Packaged Chocolate	9
Tea	2
2019	
Bakery	3
Total	

```
33a)
Top 5 Products by Profit(RANKX) =
Var ProductsRank =
RANKX(
ALL(
    'Product Lookup'[product]
),
[Profit]
)

VAR Top5Products =
IF(
    ProductsRank <= 5,
    [Profit],
    BLANK()
)
RETURN
Top5Products
```



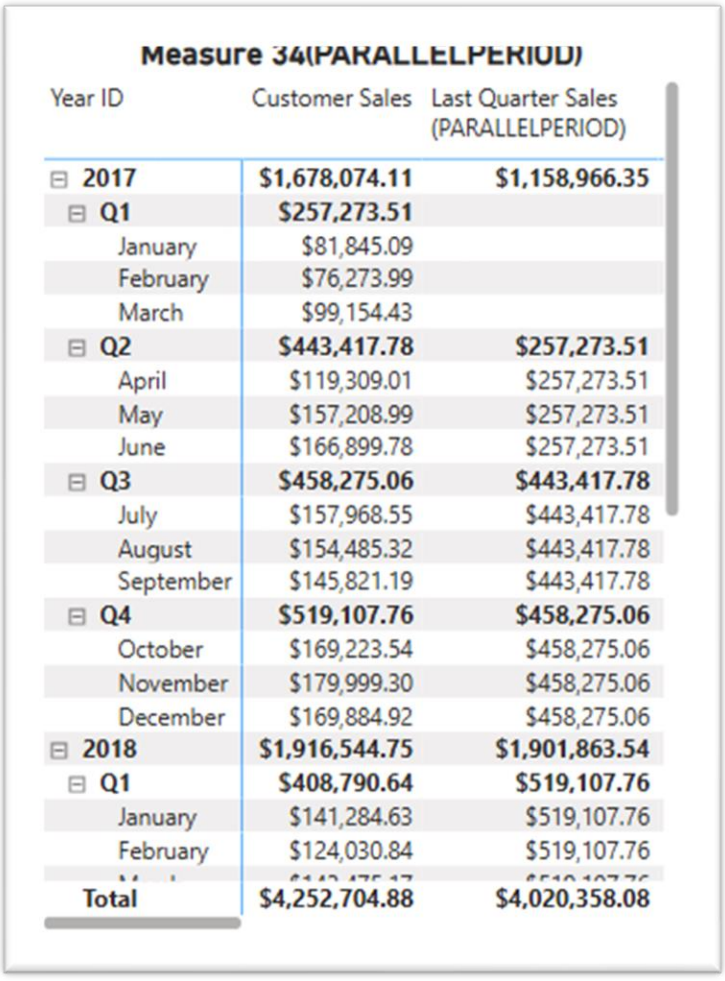
Measure 33a(RANKX)

Year ID	Top 5 Products by Profit(RANKX)
2017	**1,242,522.69**
Cappuccino Lg	32,436.00
Dark chocolate Lg	37,469.25
Latte Rg	34,064.81
Morning Sunrise Chai Lg	31,410.00
Sustainably Grown Organic Lg	38,108.06
2018	**1,419,047.46**
Cappuccino Lg	37,309.69
Dark chocolate Lg	42,788.25
Latte Rg	39,101.06
Morning Sunrise Chai Lg	36,825.00
Sustainably Grown Organic Lg	43,797.38
2019	**487,302.41**
Cappuccino Lg	12,606.56
Dark chocolate Lg	14,691.38
Latte Rg	13,346.06
Morning Sunrise Chai Lg	12,576.00
Sustainably Grown Organic Lg	14,916.19
Total	**3,148,872.56**

Time Intelligence

34)
Last Quarter Sales (PARALLELPERIOD) =

```
CALCULATE(
    [Customer Sales],
    PARALLELPERIOD(
        'Calendar'[Transaction Date],
        -1,
        QUARTER
    )
)
```



Measure 34(PARALLELPERIOD)

Year ID	Customer Sales	Last Quarter Sales (PARALLELPERIOD)
⊟ 2017	$1,678,074.11	$1,158,966.35
⊟ Q1	$257,273.51	
January	$81,845.09	
February	$76,273.99	
March	$99,154.43	
⊟ Q2	$443,417.78	$257,273.51
April	$119,309.01	$257,273.51
May	$157,208.99	$257,273.51
June	$166,899.78	$257,273.51
⊟ Q3	$458,275.06	$443,417.78
July	$157,968.55	$443,417.78
August	$154,485.32	$443,417.78
September	$145,821.19	$443,417.78
⊟ Q4	$519,107.76	$458,275.06
October	$169,223.54	$458,275.06
November	$179,999.30	$458,275.06
December	$169,884.92	$458,275.06
⊟ 2018	$1,916,544.75	$1,901,863.54
⊟ Q1	$408,790.64	$519,107.76
January	$141,284.63	$519,107.76
February	$124,030.84	$519,107.76
Total	$4,252,704.88	$4,020,358.08

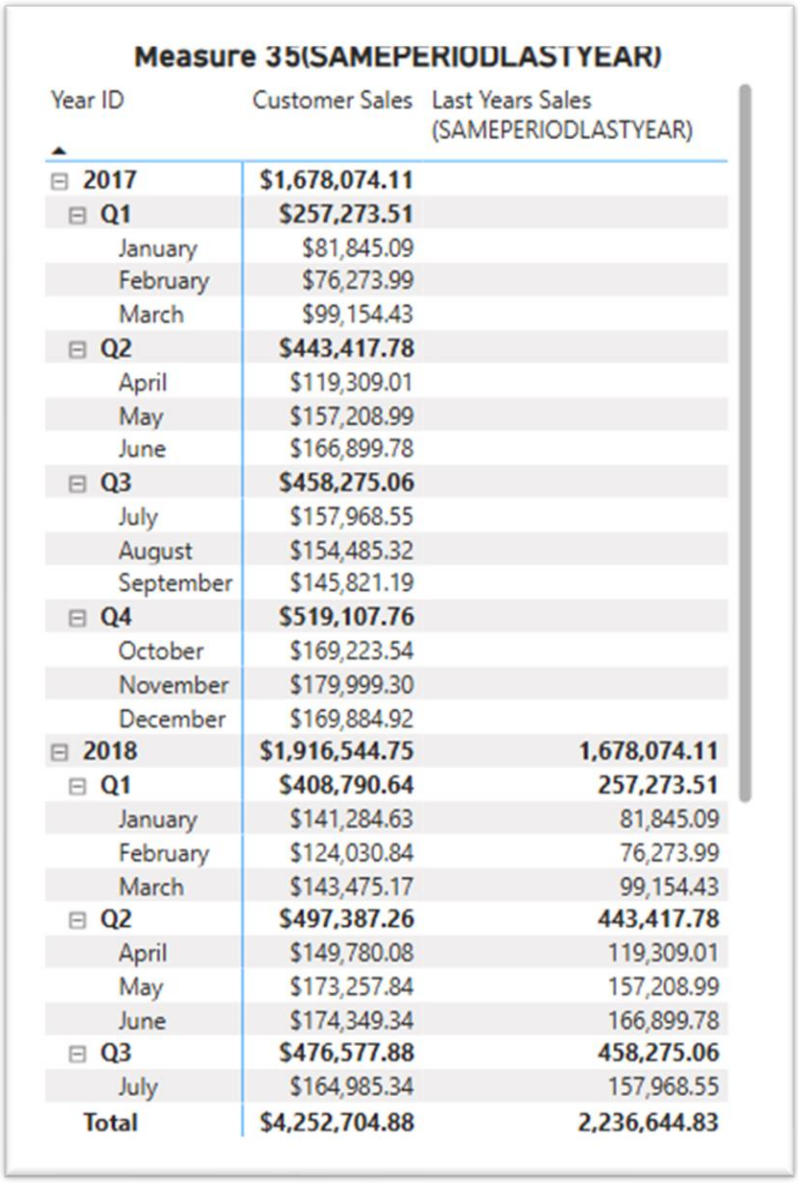
34a)–same as PARALLELPERIOD Calculations but handles total differently

```
Last Quarter Sales (PREVIOUS QUARTER) =
CALCULATE(
    [Customer Sales],
    PREVIOUSQUARTER(
        'Calendar'[Transaction Date]
    )
)
```

Measure 34(PREVIOUSQUARTER)

Year ID	Customer Sales	Last Quarter Sales(PREVIOUS QUARTER)
2017	**$1,678,074.11**	
Q1	**$257,273.51**	
January	$81,845.09	
February	$76,273.99	
March	$99,154.43	
Q2	**$443,417.78**	**$257,273.51**
April	$119,309.01	$257,273.51
May	$157,208.99	$257,273.51
June	$166,899.78	$257,273.51
Q3	**$458,275.06**	**$443,417.78**
July	$157,968.55	$443,417.78
August	$154,485.32	$443,417.78
September	$145,821.19	$443,417.78
Q4	**$519,107.76**	**$458,275.06**
October	$169,223.54	$458,275.06
November	$179,999.30	$458,275.06
December	$169,884.92	$458,275.06
2018	**$1,916,544.75**	**$519,107.76**
Q1	**$408,790.64**	**$519,107.76**
January	$141,284.63	$519,107.76
February	$124,020.84	$510,107.76
Total	**$4,252,704.88**	

```
35)
Last Years Sales (SAMEPERIODLASTYEAR) =
CALCULATE(
    [Customer Sales],
    SAMEPERIODLASTYEAR(
        'Calendar'[Transaction Date]
    )
)
```



Measure 35(SAMEPERIODLASTYEAR)

Year ID	Customer Sales	Last Years Sales (SAMEPERIODLASTYEAR)
2017	**$1,678,074.11**	
Q1	**$257,273.51**	
January	$81,845.09	
February	$76,273.99	
March	$99,154.43	
Q2	**$443,417.78**	
April	$119,309.01	
May	$157,208.99	
June	$166,899.78	
Q3	**$458,275.06**	
July	$157,968.55	
August	$154,485.32	
September	$145,821.19	
Q4	**$519,107.76**	
October	$169,223.54	
November	$179,999.30	
December	$169,884.92	
2018	**$1,916,544.75**	**1,678,074.11**
Q1	**$408,790.64**	**257,273.51**
January	$141,284.63	81,845.09
February	$124,030.84	76,273.99
March	$143,475.17	99,154.43
Q2	**$497,387.26**	**443,417.78**
April	$149,780.08	119,309.01
May	$173,257.84	157,208.99
June	$174,349.34	166,899.78
Q3	**$476,577.88**	**458,275.06**
July	$164,985.34	157,968.55
Total	**$4,252,704.88**	**2,236,644.83**

```
36)
Last Month Sales =
CALCULATE(
    [Customer Sales],
    PARALLELPERIOD(
        'Calendar'[Transaction Date],
        -1,
        MONTH
    ))
```

Measure 36(PARALLELPERIOD)

Year ID	Customer Sales	Last Month Sales
2017	**$1,678,074.11**	**$1,508,189.19**
January	$81,845.09	
February	$76,273.99	$81,845.09
March	$99,154.43	$76,273.99
April	$119,309.01	$99,154.43
May	$157,208.99	$119,309.01
June	$166,899.78	$157,208.99
July	$157,968.55	$166,899.78
August	$154,485.32	$157,968.55
September	$145,821.19	$154,485.32
October	$169,223.54	$145,821.19
November	$179,999.30	$169,223.54
December	$169,884.92	$179,999.30
2018	**$1,916,544.75**	**$1,911,335.03**
January	$141,284.63	$169,884.92
February	$124,030.84	$141,284.63
March	$143,475.17	$124,030.84
April	$149,780.08	$143,475.17
May	$173,257.84	$149,780.08
June	$174,349.34	$173,257.84
July	$164,985.34	$174,349.34
August	$161,108.60	$164,985.34
September	$150,483.94	$161,108.60
October	$174,293.02	$150,483.94
November	$184,401.31	$174,293.02
December	$175,094.64	$184,401.31
2019	**$658,086.02**	**$600,833.86**
January	$146,863.61	$175,094.64
February	$129,473.65	$146,863.61
Total	**$4,252,704.88**	**$4,020,358.08**

37)

```
Customer Sales % change (MoM) =
DIVIDE(
    [Customer Sales] - [Last Month Sales],
    [Last Month Sales],
    Blank()
)
```

Measure 37

Year ID	Customer Sales	Last Month Sales	Customer Sales % change (MoM)
2017	**$1,678,074.11**	**$1,508,189.19**	**11.26%**
January	$81,845.09		
February	$76,273.99	$81,845.09	-6.81%
March	$99,154.43	$76,273.99	30.00%
April	$119,309.01	$99,154.43	20.33%
May	$157,208.99	$119,309.01	31.77%
June	$166,899.78	$157,208.99	6.16%
July	$157,968.55	$166,899.78	-5.35%
August	$154,485.32	$157,968.55	-2.21%
September	$145,821.19	$154,485.32	-5.61%
October	$169,223.54	$145,821.19	16.05%
November	$179,999.30	$169,223.54	6.37%
December	$169,884.92	$179,999.30	-5.62%
2018	**$1,916,544.75**	**$1,911,335.03**	**0.27%**
January	$141,284.63	$169,884.92	-16.84%
February	$124,030.84	$141,284.63	-12.21%
March	$143,475.17	$124,030.84	15.68%
April	$149,780.08	$143,475.17	4.39%
May	$173,257.84	$149,780.08	15.67%
June	$174,349.34	$173,257.84	0.63%
July	$164,985.34	$174,349.34	-5.37%
August	$161,108.60	$164,985.34	-2.35%
September	$150,483.94	$161,108.60	-6.59%
October	$174,293.02	$150,483.94	15.82%
November	$184,401.31	$174,293.02	5.80%
December	$175,094.64	$184,401.31	-5.05%
2019	**$658,086.02**	**$600,833.86**	**9.53%**
Total	**$4,252,704.88**	**$4,020,358.08**	**5.78%**

```
38)
Customer Sales % change (YoY) =
VAR LastYearSales =
CALCULATE(
    [Customer Sales],
    SAMEPERIODLASTYEAR(
        'Calendar'[Transaction Date]
    )
)

VAR ratio =
DIVIDE(
    [Customer Sales] - LastYearSales,
    LastYearSales,
    "-"
)

RETURN
ratio
```

Measure 38

Year ID	Customer Sales	Last Years Sales (SAMEPERIODLASTYEAR)	Customer Sales % change (YoY)
2017	**$1,678,074.11**		-
January	$81,845.09		-
February	$76,273.99		-
March	$99,154.43		-
April	$119,309.01		-
May	$157,208.99		-
June	$166,899.78		-
July	$157,968.55		-
August	$154,485.32		-
September	$145,821.19		-
October	$169,223.54		-
November	$179,999.30		-
December	$169,884.92		-
2019	**$658,086.02**	**558,570.72**	**17.82%**
February	$129,473.65	124,030.84	4.39%
January	$146,863.61	141,284.63	3.95%
March	$149,401.96	143,475.17	4.13%
April	$232,346.80	149,780.08	55.13%
2018	**$1,916,544.75**	**1,678,074.11**	**14.21%**
February	$124,030.84	76,273.99	62.61%
January	$141,284.63	81,845.09	72.62%
March	$143,475.17	99,154.43	44.70%
April	$149,780.08	119,309.01	25.54%
September	$150,483.94	145,821.19	3.20%
August	$161,108.60	154,485.32	4.29%
May	$173,257.84	157,208.99	10.21%
July	$164,985.34	157,968.55	4.44%
Total	**$4,252,704.88**	**2,236,644.83**	**90.14%**

Week based calculation(4-5-4 Fiscal Calender)

```
39)
Last Week's Sales 4-5-4 (DATEADD) =
CALCULATE(
    [Customer Sales],
    DATEADD(
        '4-5-4 Calendar'[Date],
        -7,
        DAY
    )
)
```

FiscalYear	Customer Sales	Last Week's Sales 4-5-4 (DATEADD)
⊞	**$91,961.44**	
⊟ **2017**	**$1,739,514.10**	**$1,711,902.75**
1	$17,872.85	
2	$19,531.02	$17,872.85
3	$20,325.92	$19,531.02
4	$20,401.20	$20,325.92
5	$22,281.41	$20,401.20
6	$23,342.89	$22,281.41
7	$23,424.50	$23,342.89
8	$21,832.18	$23,424.50
9	$25,925.94	$21,832.18
10	$28,569.79	$25,925.94
11	$29,396.61	$28,569.79
12	$28,155.59	$29,396.61
13	$31,128.58	$28,155.59
14	$35,998.36	$31,128.58
15	$38,158.39	$35,998.36
16	$37,637.28	$38,158.39
17	$33,297.71	$37,637.28
18	$37,422.55	$33,297.71
19	$40,708.46	$37,422.55
20	$41,036.10	$40,708.46
21	$37,015.22	$41,036.10
22	$33,938.31	$37,015.22
23	$36,661.38	$33,938.31
24	$38,204.03	$36,661.38
25	$35,323.99	$38,204.03
26	$31,639.29	$35,323.99
27	$34,044.07	$31,639.29
Total	**$4,252,704.88**	**$4,160,743.44**

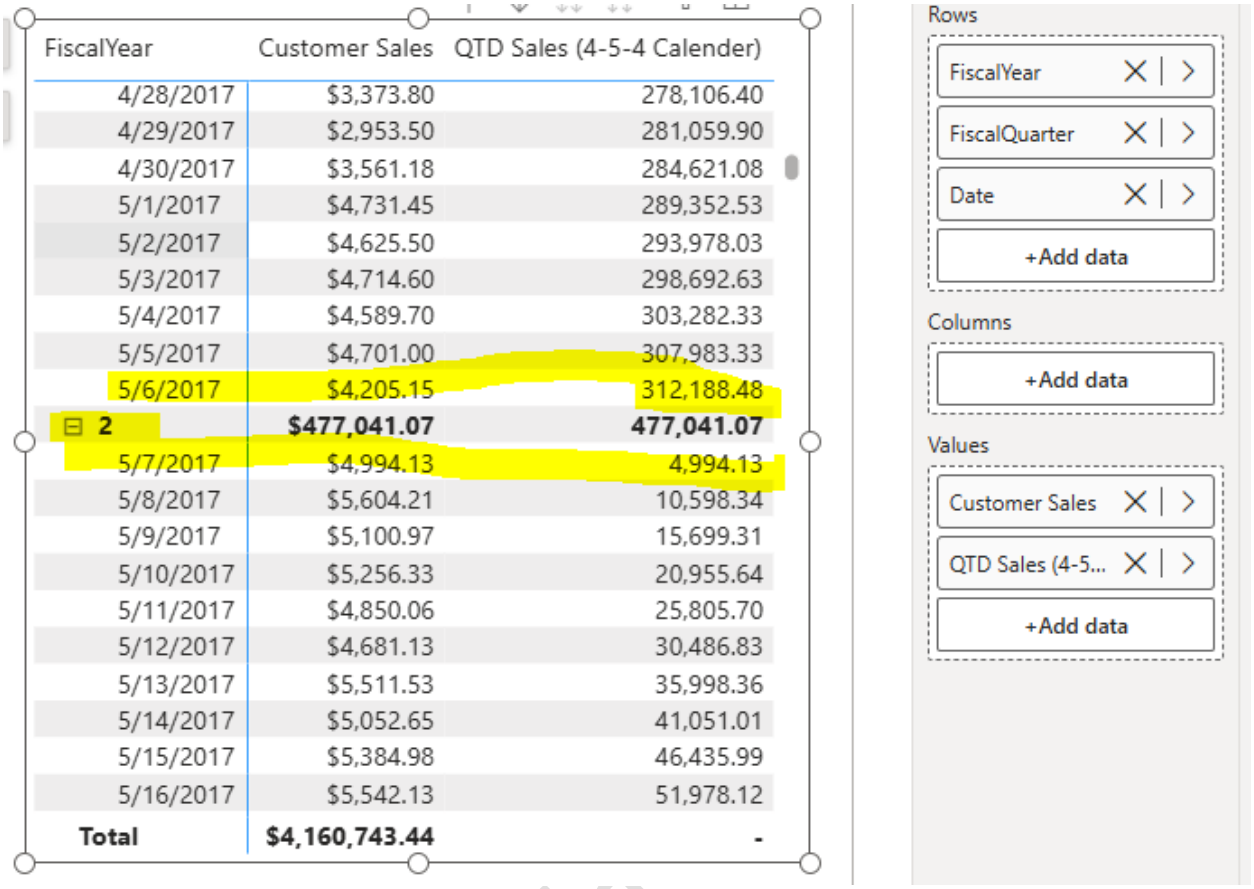
FiscalYear	Customer Sales	Last Week's Sales 4-5-4 (DATEADD)
⊞	**$91,961.44**	
⊟ **2017**	**$1,739,514.10**	**$1,711,902.75**
⊟ **1**	**$17,872.85**	
2/5/2017	$2,304.70	
2/6/2017	$2,203.40	
2/7/2017	$2,563.35	
2/8/2017	$2,762.43	
2/9/2017	$2,610.63	
2/10/2017	$2,901.60	
2/11/2017	$2,526.74	
⊟ **2**	**$19,531.02**	**$17,872.85**
2/12/2017	$2,894.00	$2,304.70
2/13/2017	$2,845.48	$2,203.40
2/14/2017	$2,673.93	$2,563.35
2/15/2017	$2,928.05	$2,762.43
2/16/2017	$3,023.33	$2,610.63
2/17/2017	$2,300.75	$2,901.60
2/18/2017	$2,865.48	$2,526.74
⊟ **3**	**$20,325.92**	**$19,531.02**
2/19/2017	$3,219.60	$2,894.00
2/20/2017	$2,883.63	$2,845.48
2/21/2017	$2,783.53	$2,673.93
2/22/2017	$2,928.70	$2,928.05
2/23/2017	$2,746.21	$3,023.33
2/24/2017	$2,940.70	$2,300.75
2/25/2017	$2,823.55	$2,865.48
⊟ **4**	**$20,401.20**	**$20,325.92**
2/26/2017	$2,956.75	$3,219.60
2/27/2017	$3,160.00	$2,883.63
Total	**$4,252,704.88**	**$4,160,743.44**

```
39a)
Customer Sales % change (WOW) =
DIVIDE(
    [Customer Sales] - [Last Week's Sales 4-5-4 (DATEADD)],
    [Last Week's Sales 4-5-4 (DATEADD)],
    "-")
```

FiscalYear	Customer Sales	Last Week's Sales 4-5-4 (DATEADD)	Customer Sales % change (WOW)
2017	**$1,739,514.10**	**$1,711,902.75**	**1.61%**
1	$17,872.85		-
2	$19,531.02	$17,872.85	9.28%
3	$20,325.92	$19,531.02	4.07%
4	$20,401.20	$20,325.92	0.37%
5	$22,281.41	$20,401.20	9.22%
6	$23,342.89	$22,281.41	4.76%
7	$23,424.50	$23,342.89	0.35%
8	$21,832.18	$23,424.50	-6.80%
9	$25,925.94	$21,832.18	18.75%
10	$28,569.79	$25,925.94	10.20%
11	$29,396.61	$28,569.79	2.89%
12	$28,155.59	$29,396.61	-4.22%
13	$31,128.58	$28,155.59	10.56%
14	$35,998.36	$31,128.58	15.64%
15	$38,158.39	$35,998.36	6.00%
16	$37,637.28	$38,158.39	-1.37%
17	$33,297.71	$37,637.28	-11.53%
Total	**$4,160,743.44**	**$4,160,743.44**	**-0.00%**

```
40)
QTD Sales (4-5-4 Calender) =

VAR MaxDate = MAX('4-5-4 Calendar'[Date])
VAR MaxFiscalQuarter = MAX('4-5-4 Calendar'[FiscalQuarterYear])

VAR QTDSales =
IF(
    HASONEVALUE(
        '4-5-4 Calendar'[FiscalQuarterYear]
    ),
    CALCULATE(
    [Customer Sales],
    '4-5-4 Calendar'[Date] <= MaxDate,
    '4-5-4 Calendar'[FiscalQuarterYear] = MaxFiscalQuarter
    ),
    "-"
)

RETURN
QTDSales
```

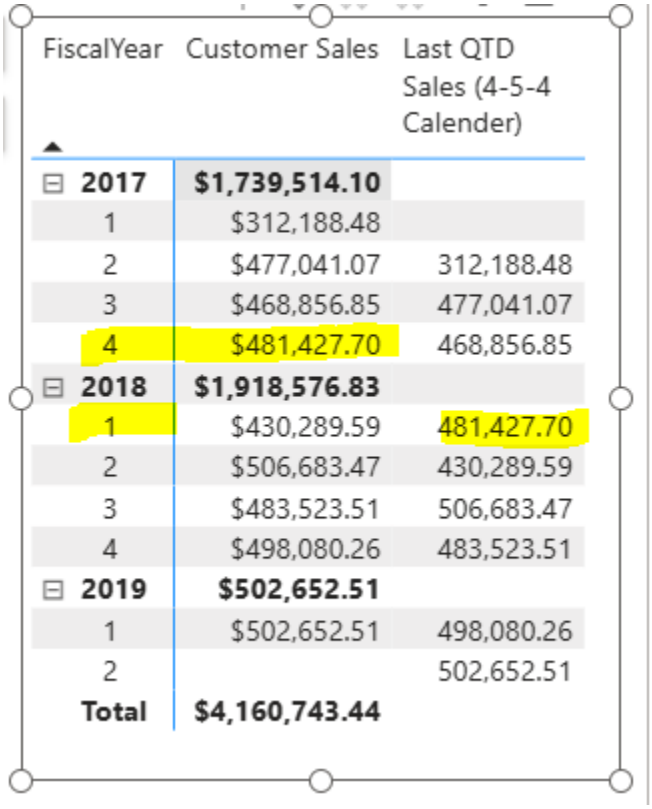
FiscalYear	Customer Sales	QTD Sales (4-5-4 Calender)
2017	**$1,739,514.10**	**-**
1	**$312,188.48**	**312,188.48**
2/5/2017	$2,304.70	2,304.70
2/6/2017	$2,203.40	4,508.10
2/7/2017	$2,563.35	7,071.45
2/8/2017	$2,762.43	9,833.88
2/9/2017	$2,610.63	12,444.51
2/10/2017	$2,901.60	15,346.11
2/11/2017	$2,526.74	17,872.85
2/12/2017	$2,894.00	20,766.85
2/13/2017	$2,845.48	23,612.33
2/14/2017	$2,673.93	26,286.26
2/15/2017	$2,928.05	29,214.31
2/16/2017	$3,023.33	32,237.64
2/17/2017	$2,300.75	34,538.39
2/18/2017	$2,865.48	37,403.87
2/19/2017	$3,219.60	40,623.47
2/20/2017	$2,883.63	43,507.10
2/21/2017	$2,783.53	46,290.63
2/22/2017	$2,928.70	49,219.33
Total	**$4,160,743.44**	**-**



FiscalYear	Customer Sales	QTD Sales (4-5-4 Calender)
4/28/2017	$3,373.80	278,106.40
4/29/2017	$2,953.50	281,059.90
4/30/2017	$3,561.18	284,621.08
5/1/2017	$4,731.45	289,352.53
5/2/2017	$4,625.50	293,978.03
5/3/2017	$4,714.60	298,692.63
5/4/2017	$4,589.70	303,282.33
5/5/2017	$4,701.00	307,983.33
5/6/2017	$4,205.15	312,188.48
2	**$477,041.07**	**477,041.07**
5/7/2017	$4,994.13	4,994.13
5/8/2017	$5,604.21	10,598.34
5/9/2017	$5,100.97	15,699.31
5/10/2017	$5,256.33	20,955.64
5/11/2017	$4,850.06	25,805.70
5/12/2017	$4,681.13	30,486.83
5/13/2017	$5,511.53	35,998.36
5/14/2017	$5,052.65	41,051.01
5/15/2017	$5,384.98	46,435.99
5/16/2017	$5,542.13	51,978.12
Total	**$4,160,743.44**	**-**

41)
Last QTD Sales (4-5-4 Calender) =

```
VAR LastPeriod =
CALCULATE(
    [Customer Sales],
    FILTER(
        ALL(
            '4-5-4 Calendar'
        ),
        IF(
            SELECTEDVALUE('4-5-4 Calendar'[FiscalQuarter]) = 1,
            '4-5-4 Calendar'[FiscalQuarter] = 4 && '4-5-4 Calendar'[FiscalYear] = SELECTEDVALUE('4-5-4 Calendar'[FiscalYear])-1,
            '4-5-4 Calendar'[FiscalYear] = SELECTEDVALUE('4-5-4 Calendar'[FiscalYear]) && '4-5-4 Calendar'[FiscalQuarter] = SELECTEDVALUE('4-5-4 Calendar'[FiscalQuarter]) -1
        )
    )
)

RETURN
LastPeriod
```

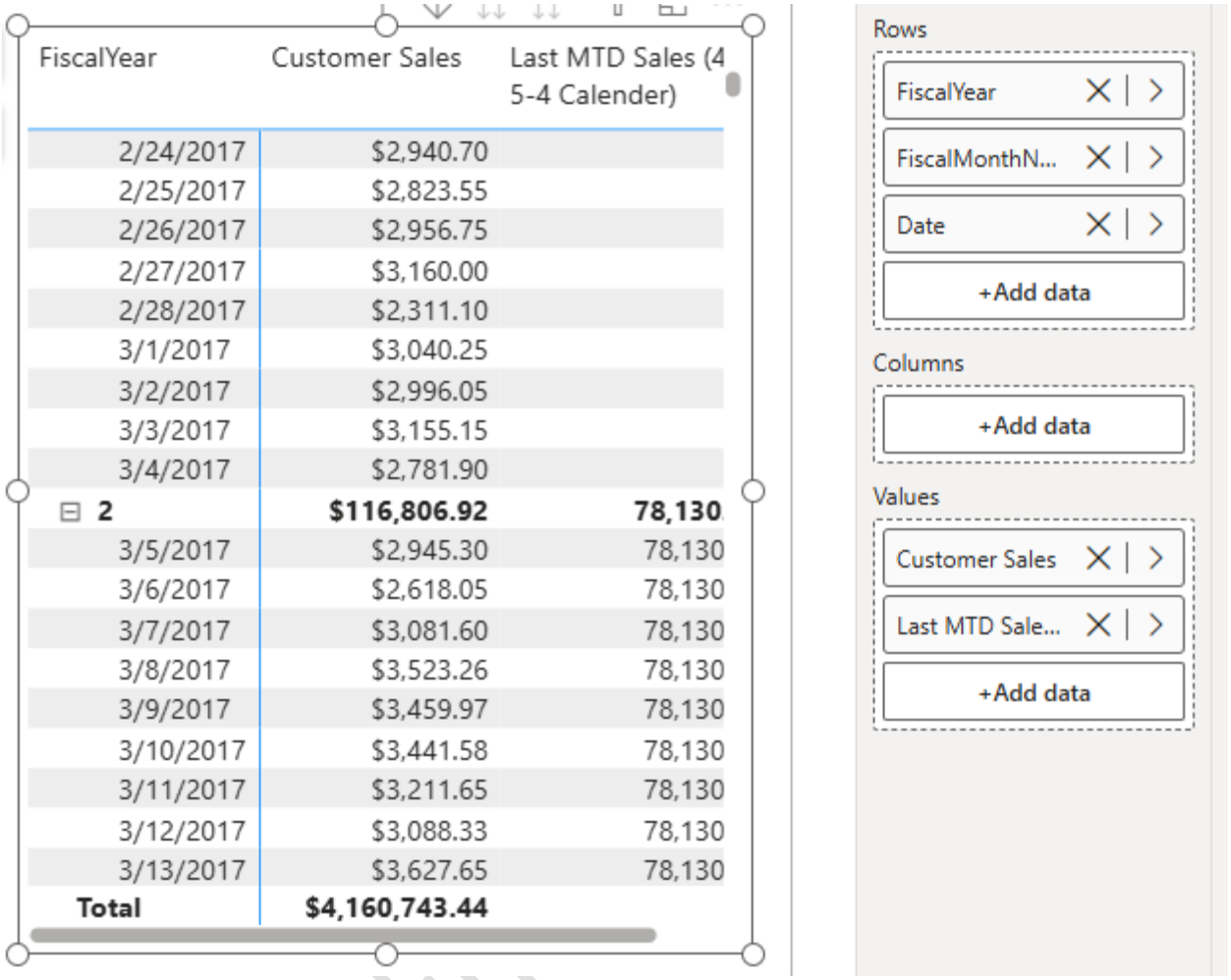


FiscalYear	Customer Sales	Last QTD Sales (4-5-4 Calender)
2017	**$1,739,514.10**	
1	$312,188.48	
2	$477,041.07	312,188.48
3	$468,856.85	477,041.07
4	$481,427.70	468,856.85
2018	**$1,918,576.83**	
1	$430,289.59	481,427.70
2	$506,683.47	430,289.59
3	$483,523.51	506,683.47
4	$498,080.26	483,523.51
2019	**$502,652.51**	
1	$502,652.51	498,080.26
2		502,652.51
Total	**$4,160,743.44**	

41a)

```
Last MTD Sales (4-5-4 Calender) =
CALCULATE(
    [Customer Sales],
    FILTER(
        ALL(
            '4-5-4 Calendar'
        ),
        IF(
            SELECTEDVALUE('4-5-4 Calendar'[FiscalMonthNumber]) = 1,
            '4-5-4 Calendar'[FiscalMonthNumber] = 12 && '4-5-4 Calendar'[FiscalYear] =
SELECTEDVALUE('4-5-4 Calendar'[FiscalYear]) -1,
            '4-5-4 Calendar'[FiscalYear] = SELECTEDVALUE('4-5-4 Calendar'[FiscalYear]) && '4-
5-4 Calendar'[FiscalMonthNumber] = SELECTEDVALUE('4-5-4 Calendar'[FiscalMonthNumber]) - 1
        )
    )
)
```

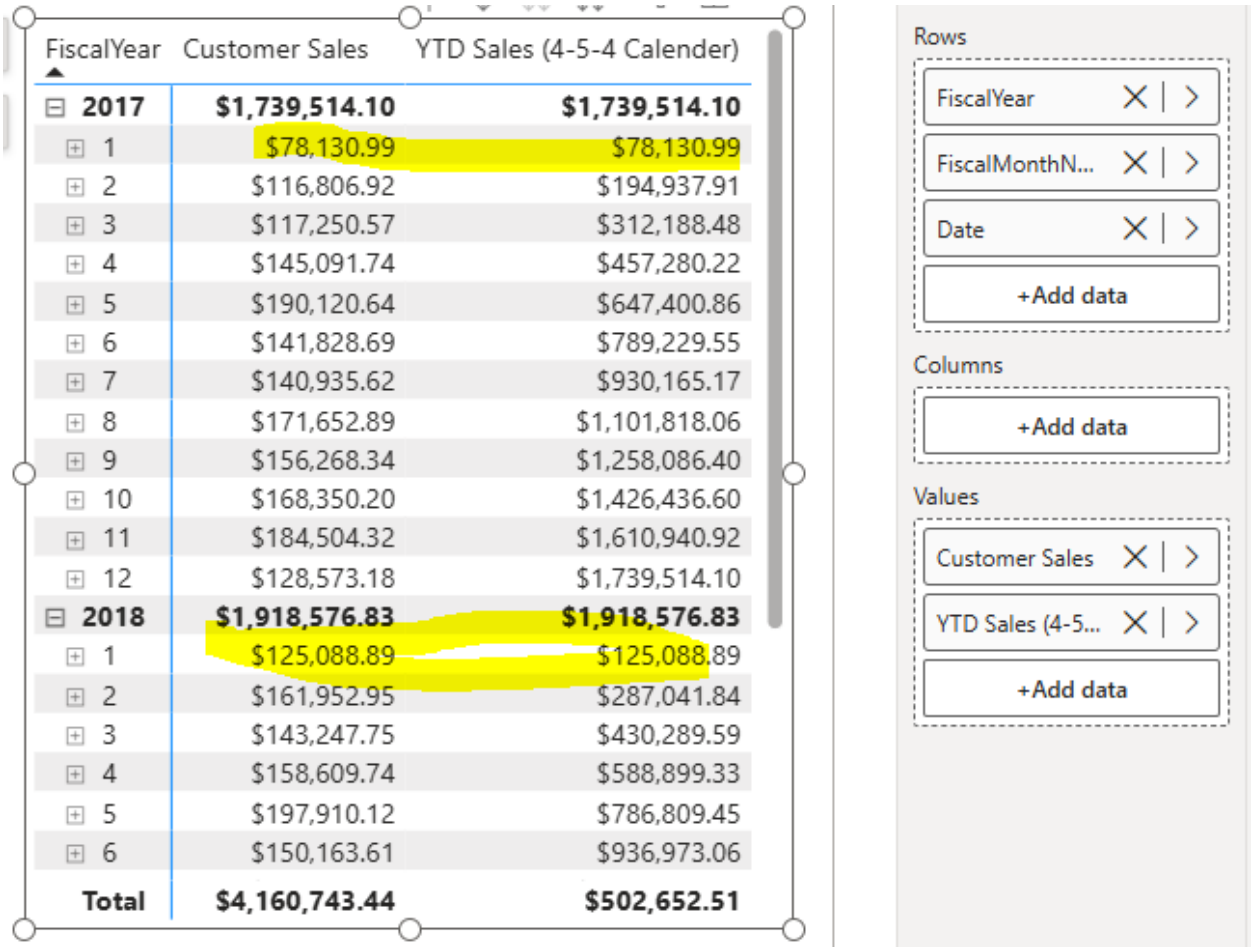
FiscalYear	Customer Sales	Last MTD Sales (4-5-4 Calender)
2017	**$1,739,514.10**	
1	$78,130.99	
2	$116,806.92	78,130.99
3	$117,250.57	116,806.92
4	$145,091.74	117,250.57
5	$190,120.64	145,091.74
6	$141,828.69	190,120.64
7	$140,935.62	141,828.69
8	$171,652.89	140,935.62
9	$156,268.34	171,652.89
10	$168,350.20	156,268.34
11	$184,504.32	168,350.20
12	$128,573.18	184,504.32
2018	**$1,918,576.83**	
1	$125,088.89	128,573.18
2	$161,952.95	125,088.89
3	$143,247.75	161,952.95
4	$158,609.74	143,247.75
5	$197,910.12	158,609.74
Total	**$4,160,743.44**	



FiscalYear	Customer Sales	Last MTD Sales (4 5-4 Calender)
2/24/2017	$2,940.70	
2/25/2017	$2,823.55	
2/26/2017	$2,956.75	
2/27/2017	$3,160.00	
2/28/2017	$2,311.10	
3/1/2017	$3,040.25	
3/2/2017	$2,996.05	
3/3/2017	$3,155.15	
3/4/2017	$2,781.90	
2	**$116,806.92**	**78,130**
3/5/2017	$2,945.30	78,130
3/6/2017	$2,618.05	78,130
3/7/2017	$3,081.60	78,130
3/8/2017	$3,523.26	78,130
3/9/2017	$3,459.97	78,130
3/10/2017	$3,441.58	78,130
3/11/2017	$3,211.65	78,130
3/12/2017	$3,088.33	78,130
3/13/2017	$3,627.65	78,130
Total	**$4,160,743.44**	

42)

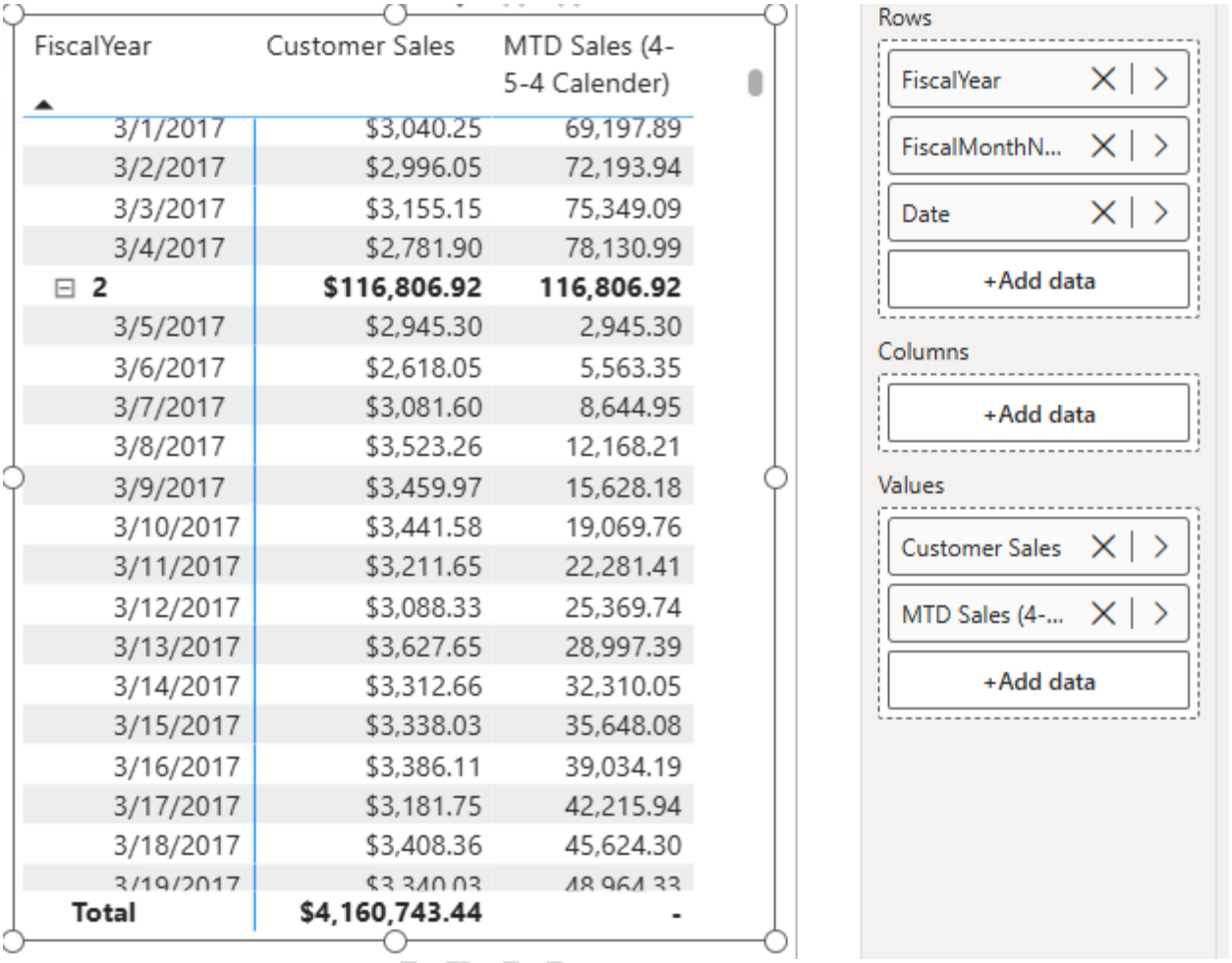
```
YTD Sales (4-5-4 Calender) =

var maxdate = MAX('4-5-4 Calendar'[Date])
VAR maxperiod = MAX('4-5-4 Calendar'[FiscalYear])
VAR MaxSaledDate = MAX('Sales by Store'[transaction date])

VAR YTDSales =
CALCULATE(
    [Customer Sales],
    '4-5-4 Calendar'[Date] <= maxdate,
    '4-5-4 Calendar'[FiscalYear] = maxperiod,
    'Calendar'[Transaction Date] <= MaxSaledDate
)

RETURN
YTDSales
```



FiscalYear	Customer Sales	YTD Sales (4-5-4 Calender)
2017	**$1,739,514.10**	**$1,739,514.10**
1	$78,130.99	$78,130.99
2	$116,806.92	$194,937.91
3	$117,250.57	$312,188.48
4	$145,091.74	$457,280.22
5	$190,120.64	$647,400.86
6	$141,828.69	$789,229.55
7	$140,935.62	$930,165.17
8	$171,652.89	$1,101,818.06
9	$156,268.34	$1,258,086.40
10	$168,350.20	$1,426,436.60
11	$184,504.32	$1,610,940.92
12	$128,573.18	$1,739,514.10
2018	**$1,918,576.83**	**$1,918,576.83**
1	$125,088.89	$125,088.89
2	$161,952.95	$287,041.84
3	$143,247.75	$430,289.59
4	$158,609.74	$588,899.33
5	$197,910.12	$786,809.45
6	$150,163.61	$936,973.06
Total	**$4,160,743.44**	**$502,652.51**

```
43)
MTD Sales (4-5-4 Calender) =
VAR Maxdate = MAX('4-5-4 Calendar'[Date])
VAR MaxPeriod = MAX('4-5-4 Calendar'[FiscalMonthYear])
VAR MTDSales =
IF(
    HASONEVALUE(
        '4-5-4 Calendar'[FiscalMonthName]),
        CALCULATE(
            [Customer Sales],
            '4-5-4 Calendar'[Date] <= Maxdate,
            '4-5-4 Calendar'[FiscalMonthYear] = MaxPeriod
        ),
        "-"
)

RETURN
MTDSales
```

FiscalYear	Customer Sales	MTD Sales (4-5-4 Calender)
2017	**$1,739,514.10**	**-**
1	**$78,130.99**	**78,130.99**
2/5/2017	$2,304.70	2,304.70
2/6/2017	$2,203.40	4,508.10
2/7/2017	$2,563.35	7,071.45
2/8/2017	$2,762.43	9,833.88
2/9/2017	$2,610.63	12,444.51
2/10/2017	$2,901.60	15,346.11
2/11/2017	$2,526.74	17,872.85
2/12/2017	$2,894.00	20,766.85
2/13/2017	$2,845.48	23,612.33
2/14/2017	$2,673.93	26,286.26
2/15/2017	$2,928.05	29,214.31
2/16/2017	$3,023.33	32,237.64
2/17/2017	$2,300.75	34,538.39
2/18/2017	$2,865.48	37,403.87
2/19/2017	$3,219.60	40,623.47
2/20/2017	$2,883.63	43,507.10
2/21/2017	$2,783.53	46,290.63
Total	**$4,160,743.44**	**-**



FiscalYear	Customer Sales	MTD Sales (4-5-4 Calender)
3/1/2017	$3,040.25	69,197.89
3/2/2017	$2,996.05	72,193.94
3/3/2017	$3,155.15	75,349.09
3/4/2017	$2,781.90	78,130.99
2	**$116,806.92**	**116,806.92**
3/5/2017	$2,945.30	2,945.30
3/6/2017	$2,618.05	5,563.35
3/7/2017	$3,081.60	8,644.95
3/8/2017	$3,523.26	12,168.21
3/9/2017	$3,459.97	15,628.18
3/10/2017	$3,441.58	19,069.76
3/11/2017	$3,211.65	22,281.41
3/12/2017	$3,088.33	25,369.74
3/13/2017	$3,627.65	28,997.39
3/14/2017	$3,312.66	32,310.05
3/15/2017	$3,338.03	35,648.08
3/16/2017	$3,386.11	39,034.19
3/17/2017	$3,181.75	42,215.94
3/18/2017	$3,408.36	45,624.30
3/19/2017	$3,340.03	48,964.33
Total	**$4,160,743.44**	**-**

CALCULATED COLUMN

1)

```
Time Group =

FLOOR(
    'Sales by Store'[transaction time],
    "1:0"       -- 1 hour multiple
)
```

transaction time	Time Group	T
6:45:47 AM	6:00:00 AM	
9:37:14 AM	9:00:00 AM	
9:28:07 AM	9:00:00 AM	
7:33:45 AM	7:00:00 AM	
6:45:47 AM	6:00:00 AM	
7:33:45 AM	7:00:00 AM	
8:58:59 AM	8:00:00 AM	
7:33:45 AM	7:00:00 AM	
9:12:37 AM	9:00:00 AM	
9:05:44 AM	9:00:00 AM	
8:58:59 AM	8:00:00 AM	
11:16:10 AM	11:00:00 AM	
6:34:09 AM	6:00:00 AM	
9:28:07 AM	9:00:00 AM	
8:39:16 AM	8:00:00 AM	
9:40:46 AM	9:00:00 AM	
9:05:44 AM	9:00:00 AM	
9:37:14 AM	9:00:00 AM	
6:47:55 AM	6:00:00 AM	
6:34:09 AM	6:00:00 AM	
9:12:37 AM	9:00:00 AM	
6:45:47 AM	6:00:00 AM	
7:33:45 AM	7:00:00 AM	
6:58:10 AM	6:00:00 AM	
8:17:36 AM	8:00:00 AM	
9:05:44 AM	9:00:00 AM	
8:58:59 AM	8:00:00 AM	
11:16:10 AM	11:00:00 AM	
8:39:16 AM	8:00:00 AM	
9:28:07 AM	9:00:00 AM	
6:45:47 AM	6:00:00 AM	
7:33:45 AM	7:00:00 AM	
6:58:10 AM	6:00:00 AM	
8:17:36 AM	8:00:00 AM	
9:37:14 AM	9:00:00 AM	
11:16:10 AM	11:00:00 AM	
6:34:09 AM	6:00:00 AM	

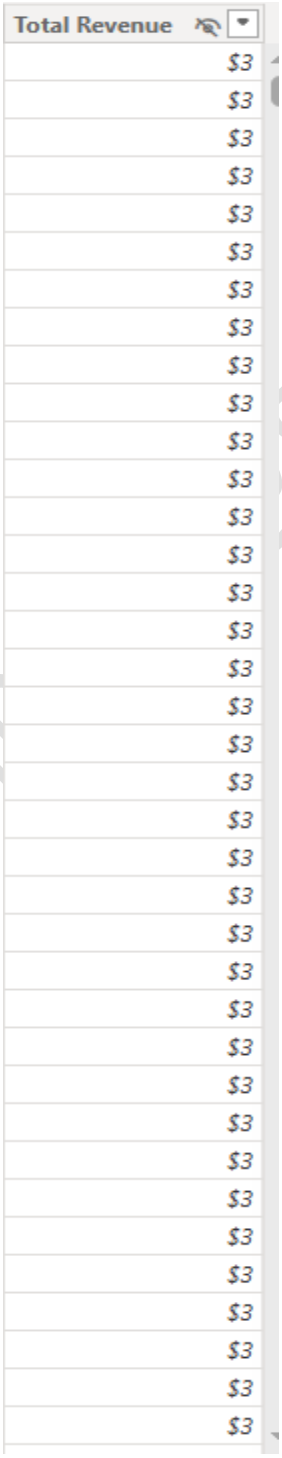
2)

```
Customer Age =
var age =
MROUND(DATEDIFF(
    Customer[birthdate],
    TODAY(),
    DAY
)/365.25, -- 365.25 because of leap year
1)
```

birthdate	Customer Age
5/13/1950	75
6/29/1950	75
8/14/1950	75
9/30/1950	75
11/16/1950	74
1/1/1951	74
2/17/1951	74
4/4/1951	74
5/21/1951	74
7/7/1951	74
8/22/1951	74
10/8/1951	74
11/23/1951	73
1/9/1952	73
2/25/1952	73
4/11/1952	73
5/28/1952	73
7/13/1952	73
8/29/1952	73
10/15/1952	73
11/30/1952	72
1/16/1953	72
3/3/1953	72
4/19/1953	72
6/5/1953	72
7/21/1953	72
9/6/1953	72
10/22/1953	72
12/8/1953	71
1/24/1954	71
3/11/1954	71
4/27/1954	71
6/12/1954	71
7/29/1954	71
9/14/1954	71
10/30/1954	71
12/16/1954	70
1/31/1955	70

3)

```
Total Revenue =

CURRENCY(
    'Sales by Store'[quantity sold] * 'Sales by Store'[unit price]
) -- Demonstration of CURRENCY function
```



4) Transaction Date (Formatted) =

```
FORMAT(
    'Calendar'[Transaction Date],
    "YYYY-mm-dd"
)
```

Transaction Date	Transaction Date (Formatted)
7/1/2017	2017-07-01
7/2/2017	2017-07-02
7/3/2017	2017-07-03
7/4/2017	2017-07-04
7/5/2017	2017-07-05
7/6/2017	2017-07-06
7/7/2017	2017-07-07
7/8/2017	2017-07-08
7/9/2017	2017-07-09
7/10/2017	2017-07-10
7/11/2017	2017-07-11
7/12/2017	2017-07-12
7/13/2017	2017-07-13
7/14/2017	2017-07-14
7/15/2017	2017-07-15
7/16/2017	2017-07-16
7/17/2017	2017-07-17
7/18/2017	2017-07-18
7/19/2017	2017-07-19
7/20/2017	2017-07-20
7/21/2017	2017-07-21
7/22/2017	2017-07-22
7/23/2017	2017-07-23
7/24/2017	2017-07-24
7/25/2017	2017-07-25
7/26/2017	2017-07-26
7/27/2017	2017-07-27
7/28/2017	2017-07-28
7/29/2017	2017-07-29
7/30/2017	2017-07-30
7/31/2017	2017-07-31
8/1/2017	2017-08-01
8/2/2017	2017-08-02
8/3/2017	2017-08-03
8/4/2017	2017-08-04
8/5/2017	2017-08-05
8/6/2017	2017-08-06

5)

```
Year Half =
SWITCH(
    'Calendar'[Month ID],
    1,"1H",
    2,"1H",
    3,"1H",
    4,"1H",
    5,"1H",
    6,"1H",
    "2H"
)
```

Month ID
7
7
7
7
7
7
7
7
7
7
7
7
7
7
7
7
7
7
7
7
7
7
7
7
7
7
7
7
7
7
7
8
8
8
8
8
8

Year Half
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H
2H

6)

```
Quarter-Year(Alternative) =
VAR Q1 = 'Calendar'[Month ID] IN {1,2,3}
VAR Q2 = 'Calendar'[Month ID] IN {4,5,6}
VAR Q3 = 'Calendar'[Month ID] IN {7,8,9}
VAR Q4 = 'Calendar'[Month ID] IN {10,11,12}

RETURN
SWITCH(
    TRUE(),
    Q1, "Q1" & "-" & 'Calendar'[Year ID],
    Q2, "Q2" & "-" & 'Calendar'[Year ID],
    Q3, "Q3" & "-" & 'Calendar'[Year ID],
    Q4, "Q4" & "-" & 'Calendar'[Year ID],
    "-"
)
```

Quarter-Year(Alternative)
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017
Q3-2017

Quarter-Year
Q2 -2017
Q2 -2017
Q2 -2017
Q2 -2017
Q2 -2017
Q2 -2017
Q2 -2017
Q2 -2017
Q2 -2017
Q2 -2017
Q2 -2017
Q2 -2017
Q2 -2017
Q2 -2017
Q2 -2017
Q2 -2017
Q2 -2017
Q2 -2017
Q2 -2017
Q2 -2017
Q2 -2017
Q4 -2017
Q4 -2017
Q4 -2017
Q4 -2017
Q4 -2017
Q4 -2017
Q4 -2017
Q4 -2017
Q4 -2017
Q4 -2017
Q4 -2017
Q4 -2017
Q4 -2017
Q4 -2017
Q4 -2017
Q4 -2017

Email: Imamuddinriad@gmail.com

Transition of Row context into Filter context. The way is using CALCULATE function.

```
7)This formula is evaluated in a calculated column, not as measure. By
default, measure evaluates filter context whether CALCULATE () is
being used or not.

Sum of Quantity (filter context - CALCULATE) =
CALCULATE(
    SUM(
        'Sales by Store'[quantity sold]
    )
)
```

Sum of Quantity (filter context - CALCULATE)
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1

RELATIONSHIP FUNCION

12)
Numbers of Food items bakced(Product Lookup) =

```
SUMX(
    RELATEDTABLE(
        'Food Inventory'
    ),
    'Food Inventory'[quantity start of day]
)
```

Numbers of Food items bakced
45198
45198
45630
120480
44658
45360
45360
45630
120480
44658
45630

```
13)
Customers Label (High\Low) =

VAR TotalSales =
SUMX(
    'Sales by Store',
    'Sales by Store'[quantity sold] * 'Sales by Store'[unit price]
)

VAR TotalCustomers =
COUNTROWS(
    DISTINCT(
    'Sales by Store'[customer id]
    )
)

VAR AVGSales =
DIVIDE(
    TotalSales,
    TotalCustomers
)

VAR IndividualSales =
SUMX(
    RELATEDTABLE(
        'Sales by Store'
    ),
    'Sales by Store'[quantity sold] *
'Sales by Store'[unit price]
)

VAR Labels =
IF(
    IndividualSales > AVGSales,
    "High",
    "Low"
)


RETURN
Labels
```

Customers Label(High\Low)
High
High
High
High
High
High
High
High
High
High
High
High
High
High
High
High
High
High
High
High
High
High
High
High
High
High
High
High
High
High
High
High
High
High
High
High
High
High

14)

```
WeekDay =
WEEKDAY([Transaction Date],2)
```

15)

```
Weekday name =
FORMAT('Calendar'[Transaction Date],"dddd")
```

16)

```
WeekEnd =
SWITCH(
    'Calendar'[WeekDay],
    6,"Y",
    7,"Y",
    "N"
)
```

Transaction Date
7/1/2017
7/2/2017
7/3/2017
7/4/2017
7/5/2017
7/6/2017
7/7/2017
7/8/2017
7/9/2017
7/10/2017
7/11/2017
7/12/2017
7/13/2017
7/14/2017
7/15/2017
7/16/2017
7/17/2017
7/18/2017
7/19/2017
7/20/2017
7/21/2017
7/22/2017

Quarter-Year	Year Half	Quarter-Year(Alternative)
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017
Q3 -2017	2H	Q3-2017

CALCULATED TABLES

1)

```
X-Demo Distinct =

DISTINCT(
    'Product Lookup'[product category]
)
```

product category
Coffee beans
Loose Tea
Packaged Chocolate
Coffee
Tea
Drinking Chocolate
Flavours
Bakery
Branded

2)VALUES returns a blank row if any record is in the fact tables but not in the dimension\lookup tables.

```
X-Demo VALUES =
VALUES(
    'Product Lookup'[product category]
)
```

product category
Coffee beans
Loose Tea
Packaged Chocolate
Coffee
Tea
Drinking Chocolate
Flavours
Bakery
Branded

2a)

```
X-Demo SELECTCOLUMNS =
SELECTCOLUMNS(
    Employee,
    "Employee ID", Employee[staff id],
    "Employee Full Name", Employee[first_name] & "-" &
Employee[last_name],
    "Employee For(years)", DATEDIFF(Employee[start date],TODAY(),YEAR)
)
```

Employee ID	Employee Full Name	Employee For(years)
1	Mark-Brewer	24
2	Jean-LeBean	24
3	Jamie-Toast	18
4	Chelsea-Claudia	22
5	Adam-Songs	17
6	Karen-Cupps	9
7	Kelsey-Cameron	22
8	Hamilton-Emi	20
9	Caldwell-Veda	12
10	Ima-Winifred	9
11	Ruth-Leslie	16
12	Britanni-Jorden	19
13	Berk-Derek	16

2b)

```
X-Demo SELECTCOLUMNS 2 =
SELECTCOLUMNS(
    FILTER(
        Employee,
        Employee[staff id] IN {6,16,31}
    ),
    "Manager ID" , Employee[staff id],
    "Full Name", Employee[first_name] & " " & Employee[last_name] ,
    "Store", Employee[location])
```

Manager ID	Full Name	Store
6	Karen Cupps	3
16	Darren Xu	5
31	Lisa Latte	8

3) X-Demo ADDCOLUMNS =

```
ADDCOLUMNS(
    FILTER(
        Employee,
        Employee[staff id] IN {6,16,31}
    ),
    "Full Name", Employee[first_name] & " " & Employee[last_name] ,
    "Store", Employee[location],
    "Employee Since", YEAR(Employee[start date])
)
```

Full Name	Store	staff id	first_name	last_name	position	start date	location	Employee Since
Karen Cupps	3	6	Karen	Cupps	Store Manager	7/24/2016 12:00:00 AM	3	2016
Darren Xu	5	16	Darren	Xu	Store Manager	3/30/2006 12:00:00 AM	5	2006
Lisa Latte	8	31	Lisa	Latte	Store Manager	7/2/2009 12:00:00 AM	8	2009

staff id	first_name	last_name	position	start date	location
1	Mark	Brewer	CFO	Friday, August 3, 2001	HQ
2	Jean	LeBean	CEO	Friday, August 3, 2001	HQ
3	Jamie	Toast	Head Roaster	Wednesday, October 24, 2007	WH
4	Chelsea	Claudia	Roaster	Thursday, July 3, 2003	WH
5	Adam	Songs	Head Barista	Wednesday, April 2, 2008	WH
6	Karen	Cupps	Store Manager	Sunday, July 24, 2016	3
7	Kelsey	Cameron	Coffee Wrangler	Saturday, October 18, 2003	3
8	Hamilton	Emi	Coffee Wrangler	Wednesday, February 9, 2005	3

```
4)
X-Unsold Pestries (SUMMARIZE) =
SUMMARIZE(
    FILTER(
        'Food Inventory',
        'Food Inventory'[quantity start of day] <> 'Food
Inventory'[quantity sold]
    ),
    'Food Inventory'[transaction date],
    'Food Inventory'[store id],
    'Food Inventory'[quantity start of day],
    'Food Inventory'[quantity sold],
    'Product Lookup'[product],
    'Product Lookup'[current_retail_price]
)
```

transaction date	store id	quantity start of day	quantity sold	current_retail_price	product
1/1/2017	8	18	3	3.5	Chocolate Chip Biscotti
1/9/2017	8	18	3	3.5	Chocolate Chip Biscotti
1/18/2017	8	18	3	3.5	Chocolate Chip Biscotti
1/19/2017	8	18	3	3.5	Chocolate Chip Biscotti
1/23/2017	8	18	3	3.5	Chocolate Chip Biscotti
2/4/2017	8	18	3	3.5	Chocolate Chip Biscotti
2/10/2017	8	18	3	3.5	Chocolate Chip Biscotti
2/12/2017	8	18	3	3.5	Chocolate Chip Biscotti
2/25/2017	8	18	3	3.5	Chocolate Chip Biscotti
3/18/2017	8	18	3	3.5	Chocolate Chip Biscotti
3/20/2017	8	18	3	3.5	Chocolate Chip Biscotti
3/21/2017	8	18	3	3.5	Chocolate Chip Biscotti
3/23/2017	8	18	3	3.5	Chocolate Chip Biscotti
3/25/2017	8	18	3	3.5	Chocolate Chip Biscotti
3/26/2017	8	18	3	3.5	Chocolate Chip Biscotti
4/2/2017	8	18	3	3.5	Chocolate Chip Biscotti

Measures based on "X-Unsold Pestries (SUMMARIZE)" table only

```
--

a)

Quantity Sold(Unsold Pestries) =

SUM(
    'X-Unsold Pestries (SUMMARIZE)'[quantity sold]
)
--

b)

Quantity Unsold(Unsold Pestries ) =
SUMX(
    'X-Unsold Pestries (SUMMARIZE)',
    'X-Unsold Pestries (SUMMARIZE)'[quantity start of day] - 'X-Unsold
Pestries (SUMMARIZE)'[quantity sold]
)
--
c)

Lost Revenue =
SUMX(
    'X-Unsold Pestries (SUMMARIZE)',
    'X-Unsold Pestries (SUMMARIZE)'[Quantity Unsold(Unsold Pestries )]
* 'X-Unsold Pestries (SUMMARIZE)'[current_retail_price]
)
```

Measure a, b, c

store id	Quantity Sold(Unsold Pestries)	Quantity Unsold(Unsold Pestries)	Lost Revenue
⊟ 3	**45,454**	**169934**	**$580,704.50**
Almond Croissant	3,798	10764	$40,365.00
Chocolate Chip Biscotti	3,893	11353	$39,735.50
Chocolate Croissant	6,152	9004	$33,765.00
Cranberry Scone	4,123	10979	$35,681.75
Croissant	3,557	11545	$37,521.25
Ginger Biscotti	3,868	11306	$39,571.00
Ginger Scone	4,155	35877	$116,600.25
Hazelnut Biscotti	4,004	11206	$39,221.00
Jumbo Savory Scone	4,421	10843	$40,661.25
Oatmeal Scone	3,916	36116	$108,348.00
Scottish Cream Scone	3,567	10941	$49,234.50
⊟ 5	**49,016**	**167836**	**$574,161.50**
Almond Croissant	4,039	10847	$40,676.25
Chocolate Chip Biscotti	4,549	10697	$37,439.50
Chocolate Croissant	6,415	8777	$32,913.75
Total	**140,893**	**506849**	**$1,732,635.50**

Table again

5)

```
X-Demo ROW =
ROW(
    "Customer Sales", [Customer Sales],
    "Cost", [Cost],
    "Profit", [Profit], --Precalculated Measure
    "Item Ordered",
    SUMX(
        'Sales by Store',
        'Sales by Store'[quantity sold]
    )       -- or calculated instant
)
```

Customer Sales	Cost	Profit	Item Ordered
4252704.88	1103832.3195	3148872.5605	1305637

6a)

```
X-Demo DATATABLE =
DATATABLE(
    "Test", INTEGER,
    {
        {10}  --One column and One Row Table
    }
)
```

Test
10

6b)

```
X-Demo DATATABLE 2 =
DATATABLE(
    "Test", INTEGER,
    "Heads || Tails", STRING,
    "Trial", INTEGER,

    {
        { 10 , "Head", 1},    -- Row 1
        { 11, "Tail", 2} , -- Row 2
        { 12, "Head", 1} -- Row 3
    }
)
```

Test	Heads || Tails	Trial
10	Head	1
11	Tail	2
12	Head	1

7)

```
X-Demo GENERATESERIES =
GENERATESERIES(
    -50.5,
    50.5,
    10
)
```

Value
7
14
21
28
35
42
49
56
63

8a)

```
X-Demo TableConstructor =
{
    1       --Single row, single column
}
```

8b)

```
X-Demo TableConstructor 2 =

{
    (1,2,3,5)       --Single row, multiple columns
}
```

8c)

```
X-Demo TableConstructor 3 =
{
    (1),    -- Multiple Rows, single column
    (2),
    (3),
    (4)
}
```

Value
1
2
3
4

8d)

```
X-Demo TableConstructor 4 =
{
    (1,2,3,4),    -- Multiple Rows, multiple Columns
    (5,6,7,8),
    (9,10,11,20),
    (12,13,14,15)
}
```

Value1	Value2	Value3	Value4
1	2	3	4
5	6	7	8
9	10	11	20
12	13	14	15

8e)

```
X-Demo TableConstructor 5 =

{
   ("Customer 1", CURRENCY(CALCULATE([Customer Sales],'Customer'[customer_id] = 1))),
   ("Customer 2", CURRENCY(CALCULATE([Customer Sales],'Customer'[customer_id] = 2))),
   ("Customer 3", CURRENCY(CALCULATE([Customer Sales],'Customer'[customer_id] = 3))),
   ("Customer 4", CURRENCY(CALCULATE([Customer Sales],'Customer'[customer_id] = 4))),
   ("Customer 5", CURRENCY(CALCULATE([Customer Sales],'Customer'[customer_id] = 5))),
   ("Customer 6", CURRENCY(CALCULATE([Customer Sales],'Customer'[customer_id] = 6)))
}
```

Value1	Value2
Customer 1	$556.8
Customer 2	$1,598.15
Customer 3	$3,217.65
Customer 4	$500.35
Customer 5	$440
Customer 6	$826.7

9)

```
X-Demo DATATABLE (TARGET SALES April-2019) =
DATATABLE(
    "Store Id" , INTEGER,
    "Year", INTEGER,
    "Month", STRING,
    "Bean/Teas Goal", INTEGER,
    "Beverage Goal", INTEGER,
    "Food Goal", INTEGER,
    "Merchandise Goal", INTEGER,
    {
        {3,2019,"April",268,15608,1964,80},
        {5,2019,"April",277,14687,2020,91},
        {8,2019,"April",377,15011,1973,34}
    }
)
```

Store Id	Year	Month	Bean/Teas Goal	Beverage Goal	Food Goal	Merchandise Goal
3	2019	April	268	15608	1964	80
5	2019	April	277	14687	2020	91
8	2019	April	377	15011	1973	34

9a)

```
X-Demo DATATABLE (TARGET SALES March-2019) =
DATATABLE(
    "Store Id" , INTEGER,
    "Year", INTEGER,
    "Month", STRING,
    "Bean/Teas Goal", INTEGER,
    "Beverage Goal", INTEGER,
    "Food Goal", INTEGER,
    "Merchandise Goal", INTEGER,
    {
        {3,2019,"March",286,15708,1970,88},
        {5,2019,"March",287,14787,2025,87},
        {8,2019,"March",390,15111,1983,43}
    }
)
```

Store Id	Year	Month	Bean/Teas Goal	Beverage Goal	Food Goal	Merchandise Goal
3	2019	March	286	15708	1970	88
5	2019	March	287	14787	2025	87
8	2019	March	390	15111	1983	43

10)

```
X-Demo CROSSJOIN =

CROSSJOIN(
    'Product Lookup',
    'Store Lookup'
)
```

product id	product group	product category	product type	product	product_description	current cost	current wholesale price
1	Whole Bean/Teas	Coffee beans	Organic Beans	Brazilian - Organic	It's like Carnival in a cup. Clean and smooth.	3.6	14.4
2	Whole Bean/Teas	Coffee beans	House blend Beans	Our Old Time Diner Blend	Out packed blend of beans that is reminiscent of the cup of coffee you used to get at a diner.	3.6	14.4
3	Whole Bean/Teas	Coffee beans	Espresso Beans	Espresso Roast	Our house blend for a good espresso shot.	2.95	11.8
4	Whole Bean/Teas	Coffee beans	Espresso Beans	Primo Espresso Roast	Our primium single source of hand roasted beans.	4.09	16.36
5	Whole Bean/Teas	Coffee beans	Gourmet Beans	Columbian Medium Roast	A smooth cup of coffee any time of day.	3	12
6	Whole Bean/Teas	Coffee beans	Gourmet Beans	Ethiopia	From the home of coffee.	4.2	16.8
7	Whole Bean/Teas	Coffee beans	Premium Beans	Jamaican Coffee River	Ya man, it will start your day off right	3.95	15.8

current wholesale price	current_retail_price	Numbers of Food items bakced	Numbers of Food items bakced 2	store id	store type	store_address	store city	store_state_province	store_postal_code
14.4	18			2	warehouse	164-14 Jamaica Ave	Jamaica	NY	11432
14.4	18			2	warehouse	164-14 Jamaica Ave	Jamaica	NY	11432
11.8	14.75			2	warehouse	164-14 Jamaica Ave	Jamaica	NY	11432
16.36	20.45			2	warehouse	164-14 Jamaica Ave	Jamaica	NY	11432
12	15			2	warehouse	164-14 Jamaica Ave	Jamaica	NY	11432
16.8	21			2	warehouse	164-14 Jamaica Ave	Jamaica	NY	11432
15.8	19.75			2	warehouse	164-14 Jamaica Ave	Jamaica	NY	11432
36	45			2	warehouse	164-14 Jamaica Ave	Jamaica	NY	11432

store_state_province	store_postal_code	store_longitude	store_latitude
NY	11432	-73.795168	40.705226
NY	11432	-73.795168	40.705226
NY	11432	-73.795168	40.705226
NY	11432	-73.795168	40.705226
NY	11432	-73.795168	40.705226

10a)

```
X-Demo CROSSJOIN 2 =
CROSSJOIN(
    VALUES(
        'Product Lookup'[product category]
    ),      -- DAX table expression\virtual table\Calculated Table 1
    VALUES(
        'Product Lookup'[product group]
    ),       -- DAX table expression\virtual table\Calculated Table 2
    FILTER(
        VALUES(
          'Store Lookup'[store id]
        ),
        'Store Lookup'[store id] = 3
    )        -- DAX table expression\virtual table\Calculated Table 3
)
```

product category	product group	store id
Coffee beans	Whole Bean/Teas	3
Loose Tea	Whole Bean/Teas	3
Packaged Chocolate	Whole Bean/Teas	3
Coffee	Whole Bean/Teas	3
Tea	Whole Bean/Teas	3
Drinking Chocolate	Whole Bean/Teas	3
Flavours	Whole Bean/Teas	3
Bakery	Whole Bean/Teas	3
Branded	Whole Bean/Teas	3
Coffee beans	Beverages	3
Loose Tea	Beverages	3

11)

```
X-Demo UNION =
UNION(
    'X-Demo DATATABLE (TARGET SALES April-2019)',
    'X-Demo DATATABLE (TARGET SALES March-2019)'
)
```

Store Id	Year	Month	Bean/Teas Goal	Beverage Goal	Food Goal	Merchandise Goal
3	2019	March	286	15708	1970	88
5	2019	March	287	14787	2025	87
8	2019	March	390	15111	1983	43

Store Id	Year	Month	Bean/Teas Goal	Beverage Goal	Food Goal	Merchandise Goal
3	2019	April	268	15608	1964	80
5	2019	April	277	14687	2020	91
8	2019	April	377	15011	1973	34

Store Id	Year	Month	Bean/Teas Goal	Beverage Goal	Food Goal	Merchandise Goal
3	2019	March	286	15708	1970	88
5	2019	March	287	14787	2025	87
8	2019	March	390	15111	1983	43
3	2019	April	268	15608	1964	80
5	2019	April	277	14687	2020	91
8	2019	April	377	15011	1973	34

X-Demo Union

11a)

```
X-Demo UNION(Tables are in different order) =
UNION(
    'X-Demo DATATABLE (TARGET SALES March-2019)',
    SUMMARIZE(          -- The Seocnd table are not in the same order
as first table,SUMMARIZE function is used to reorder the table
        'X-Demo DATATABLE (TARGET SALES April-2019-Different Order)',
        'X-Demo DATATABLE (TARGET SALES April-2019-Different
Order)'[Store Id],
        'X-Demo DATATABLE (TARGET SALES April-2019-Different
Order)'[Year],
        'X-Demo DATATABLE (TARGET SALES April-2019-Different
Order)'[Month],
        'X-Demo DATATABLE (TARGET SALES April-2019-Different
Order)'[Bean/Teas Goal],
        'X-Demo DATATABLE (TARGET SALES April-2019-Different
Order)'[Beverage Goal],
        'X-Demo DATATABLE (TARGET SALES April-2019-Different
Order)'[Food Goal],
        'X-Demo DATATABLE (TARGET SALES April-2019-Different
Order)'[Merchandise Goal]
    )
)
```

Store Id	Year	Month	Bean/Teas Goal	Beverage Goal	Food Goal	Merchandise Goal
3	2019	March	286	15708	1970	88
5	2019	March	287	14787	2025	87
8	2019	March	390	15111	1983	43

Store Id	Year	Month	Bean/Teas Goal	Beverage Goal	Food Goal	Merchandise Goal
3	2019	April	268	15608	1964	80
5	2019	April	277	14687	2020	91
8	2019	April	377	15011	1973	34

Store Id	Year	Month	Bean/Teas Goal	Beverage Goal	Food Goal	Merchandise Goal
3	2019	March	286	15708	1970	88
5	2019	March	287	14787	2025	87
8	2019	March	390	15111	1983	43
3	2019	April	268	15608	1964	80
5	2019	April	277	14687	2020	91
8	2019	April	377	15011	1973	34

12)

```
X-Demo EXCEPT =
EXCEPT(
    Customer,
    FILTER(
        VALUES(
            Customer
        ),
        Customer[customer since] > DATE(2017,02,1)
    )
)   -- The resulting table comprises only the rows of the customers
that has not visited after 31 January 2017
    -- The EXCEPT function compares Table1 with Table2 and returns
only the rows that are present in Table1 but not in Table2.
```

customer_id	home store	customer_first-name	customer since	birthdate	gender	birth year	Customer Age	AGE	Customers Label(High\Low)
301	3	Alika Rivas	1/4/2017 12:00:00 AM	5/13/1950 12:00:00 AM	F	1950	75	74.8062970568104	1147.7
302	3	Sacha Wall	1/6/2017 12:00:00 AM	6/29/1950 12:00:00 AM	F	1950	75	74.6776180698152	1206
303	3	Raya Hampton	1/8/2017 12:00:00 AM	8/14/1950 12:00:00 AM	F	1950	75	74.5516769336071	413
304	3	Belle Reyes	1/10/2017 12:00:00 AM	9/30/1950 12:00:00 AM	F	1950	74	74.4229979466119	754.85
305	3	Brooke Munoz	1/12/2017 12:00:00 AM	11/16/1950 12:00:00 AM	F	1950	74	74.2943189596167	947.8
306	3	Tanisha Wolf	1/14/2017 12:00:00 AM	1/1/1951 12:00:00 AM	F	1951	74	74.1683778234086	1512.8
307	3	Quintessa Franklin	1/16/2017 12:00:00 AM	2/17/1951 12:00:00 AM	F	1951	74	74.0396988364134	981.000000000001
308	3	Charissa Cobb	1/18/2017 12:00:00 AM	4/4/1951 12:00:00 AM	F	1951	74	73.9137577002053	513.9
309	3	Susan Mathis	1/20/2017 12:00:00 AM	5/21/1951 12:00:00 AM	F	1951	74	73.7850787132101	593.5
310	3	Cynthia Hurley	1/22/2017 12:00:00 AM	7/7/1951 12:00:00 AM	F	1951	74	73.6563997262149	628.25
311	3	Melodie Dawson	1/24/2017 12:00:00 AM	8/22/1951 12:00:00 AM	F	1951	74	73.5304585900068	940.65
312	3	Charlotte Webb	1/26/2017 12:00:00 AM	10/8/1951 12:00:00 AM	F	1951	73	73.4017796030116	552.3
313	3	Melyssa Jones	1/28/2017 12:00:00 AM	11/23/1951 12:00:00 AM	F	1951	73	73.2758384668036	1154.95

13)

```
X-Demo INTERSECT =
INTERSECT(
    'Employee', --Table 1
    FILTER(
        Employee,
        Employee[start date] > DATE(2016,12,31)
    )   -- Table 2
)
```

staff id	first_name	last_name	position	start date	location
21	Melodie	Mercedes	Store Manager	9/29/2018 12:00:00 AM	6
23	Blythe	Arsenio	Coffee Wrangler	11/22/2018 12:00:00 AM	6
25	Aline	Melanie	Coffee Wrangler	3/14/2017 12:00:00 AM	6
45	Pandora	Neville	Coffee Wrangler	3/21/2019 12:00:00 AM	10

13a)

```
X-Demo INTERSECT 2 =
INTERSECT(
    ADDCOLUMNS(
        Employee,
        "Revenue", [Customer Sales] --Adding New Columns to the
'Employee' table
    ),
    ADDCOLUMNS(
        FILTER(
            Employee,
            Employee[start date] > DATE(2016,12,31)
        ),
        "Revenue", [Customer Sales] --Adding New Columns to the
'Employee' table
    )
)
```

staff id	first_name	last_name	position	start date	location	Revenue
21	Melodie	Mercedes	Store Manager	9/29/2018 12:00:00 AM	6	
23	Blythe	Arsenio	Coffee Wrangler	11/22/2018 12:00:00 AM	6	
25	Aline	Melanie	Coffee Wrangler	3/14/2017 12:00:00 AM	6	20517.8100000001
45	Pandora	Neville	Coffee Wrangler	3/21/2019 12:00:00 AM	10	295658.49

13b)

X-Demo INTERSECT 3 (Repeat Customer between week 45 & 46) =

```
VAR Week45Customer =
CALCULATETABLE(
    VALUES(
        'Sales by Store'[customer id]
    ),
    'Calendar'[Week ID] = 45,
    'Calendar'[Year ID] = 2018
)

VAR Week46Customer =
CALCULATETABLE(
    VALUES(
        'Sales by Store'[customer id]
    ),
    'Calendar'[Week ID] = 46,
    'Calendar'[Year ID] = 2018
)

VAR RepeatingCustomer =

INTERSECT(
    Week45Customer,
    Week46Customer
)

VAR RepeatingCustomer_Revenue_Profit =
CALCULATETABLE(
    ADDCOLUMNS(
        RepeatingCustomer,
        "Revenue" , [Customer Sales],
        "Profit" , [Profit]
    ),
    'Calendar'[Week ID] = 46,
    'Calendar'[Year ID] = 2018      /* Why Are Filters Needed When
Creating RepeatingCustomer_Revenue_Profit?
Now, when you create the RepeatingCustomer_Revenue_Profit table,
you're adding the revenue and profit for those repeating customers.
But here's the key:
•   The ADDCOLUMNS function itself doesn't automatically respect the
context set by the earlier CALCULATETABLE expressions. It works based
on the current context, but since you're recalculating the revenue and
```

```
profit, you need to reapply a filter to ensure the correct context for
these calculations.
•   In the absence of specific filters inside the CALCULATETABLE for
RepeatingCustomer_Revenue_Profit, it might attempt to calculate the
Revenue and Profit based on a broader context (for instance, other
weeks or other years), even though you have already filtered the
customer sets to Week 45 and Week 46.
 */
)

RETURN
RepeatingCustomer_Revenue_Profit
```

customer id	Revenue	Profit
5180	5	3.75
244	5	3.75
8288	5	3.75
72	5	3.75
596	5	3.75
280	5	3.75
8499	5	3.75
271	5	3.75
395	5	3.75
677	5	3.75
240	5	3.75
676	5	3.75
258	5	3.75
8274	5	3.75
8356	5	3.75

FQA

If you don't add the filter again when creating the RepeatingCustomer-Revenue-Profit table for Week 46 in 2018, the returned data might not be accurate, as you could see revenue and profit figures from the wrong week (like Week 45), leading to incorrect or misleading results.

How is it possible to grab data from another week? Didn't I filter the new table when creating the Week 45 and Week 46 customer data by filtering out only Week 45 and Week 46 in 2018? So why is it necessary to add those filters again?

Ans:

You are absolutely correct in pointing out that the `Week45Customer` and `Week46Customer` tables were already filtered to only include customers from those two weeks. So, the question is: **Why is it necessary to add the `Week ID = 46` and `Year ID = 2018` filters again when calculating revenue and profit in the `RepeatingCustomer_Revenue_Profit` table?**

Let's break this down:

What Happens When You Create `Week45Customer` and `Week46Customer`?

When you define the `Week45Customer` and `Week46Customer` tables, you're correctly filtering the data to include only customers from **Week 45** and **Week 46** of **2018**.

At this point:

- `Week45Customer` contains only customer IDs from **Week 45** of **2018**.
- `Week46Customer` contains only customer IDs from **Week 46** of **2018**.

This filtering ensures that the intersection (`INTERSECT`) only contains customers who appear in both Week 45 and Week 46. So far, you're correct in understanding that this should give you the set of "repeating customers."

Why Are Filters Needed When Creating `RepeatingCustomer_Revenue_Profit`?

Now, when you create the `RepeatingCustomer_Revenue_Profit` table, you're adding the **revenue and profit** for those repeating customers. But here's the key:

- The `ADDCOLUMNS` function itself doesn't automatically respect the context set by the earlier `CALCULATETABLE` expressions. It works based on the current context, but since you're **recalculating the revenue and profit**, you need to reapply a filter to **ensure the correct context** for these calculations.
- In the absence of specific filters inside the `CALCULATETABLE` for `RepeatingCustomer_Revenue_Profit`, it might attempt to calculate the `Revenue` and `Profit` based on a broader context (for instance, other weeks or other years), even though you have already filtered the customer sets to Week 45 and Week 46.

The Issue Without Filters

Without these filters, you're relying on the default row context for `ADDCOLUMNS`, which might still reference data from outside **Week 46 of 2018**. For example:

- If the `Customer Sales` and `Profit` measures are not specifically filtered to Week 46 within the measure itself, those values could come from a different week, even if the `RepeatingCustomer` table is already filtered for the intersected customers.
- This would lead to incorrect results because you're trying to calculate the sales and profit for **Week 46**, but the default context might pull data for **other weeks** (including Week 45), causing a mismatch.

```
14)
X-Demo Calender (CALENDER) =
CALENDAR(
    43831,
    44196
)
```

Date
1/1/2020 12:00:00 AM
1/2/2020 12:00:00 AM
1/3/2020 12:00:00 AM
1/4/2020 12:00:00 AM
1/5/2020 12:00:00 AM
1/6/2020 12:00:00 AM
1/7/2020 12:00:00 AM
1/8/2020 12:00:00 AM
1/9/2020 12:00:00 AM
1/10/2020 12:00:00 AM
1/11/2020 12:00:00 AM
1/12/2020 12:00:00 AM

15)

```
X-Demo Calender2 (CALENDER) =
CALENDAR(
    DATE(2020,1,1),
    DATE(2020,12,31)
)
```

Date
1/1/2020 12:00:00 AM
1/2/2020 12:00:00 AM
1/3/2020 12:00:00 AM
1/4/2020 12:00:00 AM
1/5/2020 12:00:00 AM
1/6/2020 12:00:00 AM
1/7/2020 12:00:00 AM
1/8/2020 12:00:00 AM
1/9/2020 12:00:00 AM
1/10/2020 12:00:00 AM
1/11/2020 12:00:00 AM
1/12/2020 12:00:00 AM

16)

```
X-Demo Calender3 (CALENDER) =
CALENDAR(
   DATE(YEAR(MIN('Calendar'[Transaction Date])),1,1),
   DATE(YEAR(MAX('Calendar'[Transaction Date])),12,31)
)
```

Date
1/1/2020 12:00:00 AM
1/2/2020 12:00:00 AM
1/3/2020 12:00:00 AM
1/4/2020 12:00:00 AM
1/5/2020 12:00:00 AM
1/6/2020 12:00:00 AM
1/7/2020 12:00:00 AM
1/8/2020 12:00:00 AM
1/9/2020 12:00:00 AM
1/10/2020 12:00:00 AM
1/11/2020 12:00:00 AM
1/12/2020 12:00:00 AM

17)
X-Demo Calender (CALENDERAUTO) =

```
VAR MinYear = YEAR(MIN('Calendar'[Transaction Date]))
VAR MaxYear = YEAR(MAX('Calendar'[Transaction Date]))

VAR DateTable =
ADDCOLUMNS(
    FILTER(
        CALENDARAUTO(),
        YEAR([Date]) >= MinYear &&
        YEAR([Date]) <= MaxYear
    ),
    "Year",YEAR([Date]),
    "Month No", MONTH([Date]),
    "Month Short", FORMAT([Date],"mmm"),
    "Quarter No", INT(FORMAT([Date],"q")),
    "Quarter", "Q" & INT(FORMAT([Date],"q")),
    "Week No", WEEKNUM([Date])
)

RETURN
DateTable
```

Date	Year	Month No	Month Short	Quarter No	Quarter	Week No	Weekday	Weekday name
7/1/2017 12:00:00 AM	2017	7	Jul	3	Q3	26	6	Sat
7/2/2017 12:00:00 AM	2017	7	Jul	3	Q3	27	7	Sun
7/3/2017 12:00:00 AM	2017	7	Jul	3	Q3	27	1	Mon
7/4/2017 12:00:00 AM	2017	7	Jul	3	Q3	27	2	Tue
7/5/2017 12:00:00 AM	2017	7	Jul	3	Q3	27	3	Wed
7/6/2017 12:00:00 AM	2017	7	Jul	3	Q3	27	4	Thu
7/7/2017 12:00:00 AM	2017	7	Jul	3	Q3	27	5	Fri
7/8/2017 12:00:00 AM	2017	7	Jul	3	Q3	27	6	Sat
7/9/2017 12:00:00 AM	2017	7	Jul	3	Q3	28	7	Sun
7/10/2017 12:00:00 AM	2017	7	Jul	3	Q3	28	1	Mon
7/11/2017 12:00:00 AM	2017	7	Jul	3	Q3	28	2	Tue
7/12/2017 12:00:00 AM	2017	7	Jul	3	Q3	28	3	Wed
7/13/2017 12:00:00 AM	2017	7	Jul	3	Q3	28	4	Thu
7/14/2017 12:00:00 AM	2017	7	Jul	3	Q3	28	5	Fri
7/15/2017 12:00:00 AM	2017	7	Jul	3	Q3	28	6	Sat
7/16/2017 12:00:00 AM	2017	7	Jul	3	Q3	29	7	Sun
7/17/2017 12:00:00 AM	2017	7	Jul	3	Q3	29	1	Mon
7/18/2017 12:00:00 AM	2017	7	Jul	3	Q3	29	2	Tue

18)

```
Top 10 Products =
TOPN(
    10,
    SUMMARIZE(
        'Sales by Store',
        'Product Lookup'[product],
        "Customer Sales",[Customer Sales]
    ),
    [Customer Sales],
    DESC
)
```

product	Customer Sales
Ethiopia Lg	91511
Cappuccino	98831.25
Morning Sunrise Chai Lg	107748
Sustainably Grown Organic Rg	99037.5
Jamaican Coffee River Lg	99333.75
Cappuccino Lg	109803
Dark chocolate Lg	126598.5
Latte Rg	115349.25
Sustainably Grown Organic Lg	129095.5
Latte	104077.5

18a)

```
Top 10 Products =
TOPN(
    10,
    'Product Lookup',
    [Customer Sales],
    DESC
)
```

product	product id	product group	product category	product type	product_description	current cost	current wholesale price	current_retail_price	Numbers of Food items bakced
Ethiopia Lg	33	Beverages	Coffee	Gourmet brewed coffee	A bold cup when you want that something extra.	0.875	1.75	3.5	
Cappuccino	40	Beverages	Coffee	Barista Espresso	You will think you are in Venice when you sip this one.	0.9375	1.875	3.75	
Morning Sunrise Chai Lg	55	Beverages	Tea	Brewed Chai tea	Face the morning after your yoga routine.	1	2	4	
Sustainably Grown Organic Rg	60	Beverages	Drinking Chocolate	Hot chocolate	Just pure notes of spice.	0.9375	1.875	3.75	
Jamaican Coffee River Lg	36	Beverages	Coffee	Premium brewed coffee	Still a front runner for good premium coffee.	0.9375	1.875	3.75	
Cappuccino Lg	41	Beverages	Coffee	Barista Espresso	You will think you are in Venice when you sip this one.	1.0625	2.125	4.25	
Dark chocolate Lg	59	Beverages	Drinking Chocolate	Hot chocolate	Slightly bitter, but still very rich.	1.125	2.25	4.5	
Latte Rg	39	Beverages	Coffee	Barista Espresso	You will think you are in Venice when you sip this one.	1.0625	2.125	4.25	
Sustainably Grown Organic Lg	61	Beverages	Drinking Chocolate	Hot chocolate	Just pure notes of spice.	1.1875	2.375	4.75	
Latte	38	Beverages	Coffee	Barista Espresso	You will think you are in Venice when you sip this one.	0.9375	1.875	3.75	

```
18b)
Top 10 Products =
TOPN(
    10,
    ADDCOLUMNS(
        VALUES(
            'Product Lookup'[product]
        ),
        "Customer Sales",[Customer Sales]
    ),
    [Customer Sales],
    DESC
)
```

Customer Sales	product
91511	Ethiopia Lg
98831.25	Cappuccino
107748	Morning Sunrise Chai Lg
99037.5	Sustainably Grown Organic Rg
99333.75	Jamaican Coffee River Lg
109803	Cappuccino Lg
126598.5	Dark chocolate Lg
115349.25	Latte Rg
129095.5	Sustainably Grown Organic Lg
104077.5	Latte

LinkedIn

Portfolio