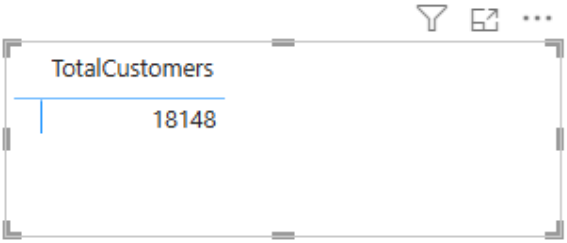
DAX Assignment

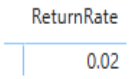
- Create a measure named Total Customers, to calculate the number of distinct AdventureWorks customers who made a transaction.

```
1 TotalCustomers = DISTINCTCOUNT('AdventureWorks Customer Lookup'[CustomerKey])
```



- Create a measure named Return Rate, defined as quantity returned divided by quantity sold.

```
ReturnRate = DIVIDE( SUM('AdventureWorks Returns Data'[ReturnQuantity]),SUM('AdventureWorks Sales Data'[OrderQuantity]), 0 )
```



a) Logical Function

- Create a calculated column in the Customer Lookup table named Customer Priority:
  → If the customer is a parent and has an annual income > $100,000, Custom Priority = Priority
  → Otherwise, Customer Priority = Standard

```
Customer Priority = IF ([TotalChildren]>0 && [AnnualIncome]>100000, "Priority", "Standard")
```

AnnualIncome	TotalChildren	EducationLevel	Occupation	HomeOwner	Domain Name	Customer Priority
100000	1	Partial College	Professional	Y	Adventure Works	Standard
100000	5	Partial College	Professional	Y	Adventure Works	Standard
100000	3	Partial College	Professional	N	Adventure Works	Standard
100000	0	Partial College	Professional	Y	Adventure Works	Standard
100000	0	Partial College	Professional	N	Adventure Works	Standard
100000	0	Partial College	Professional	Y	Adventure Works	Standard
100000	0	Partial College	Professional	Y	Adventure Works	Standard
130000	4	Partial College	Professional	N	Adventure Works	Priority
130000	3	Partial College	Professional	N	Adventure Works	Priority
130000	5	Partial College	Professional	N	Adventure Works	Priority
130000	4	Partial College	Professional	Y	Adventure Works	Priority
130000	2	Partial College	Professional	Y	Adventure Works	Priority
130000	2	Partial College	Professional	Y	Adventure Works	Priority

- Create a calculated column in the Customer Lookup table named Customer income:
  → If annual income is >= $150,000, Very High
  → If annual income is >= $100,000, High
  → If annual income is >= $50,000, Average
  → Otherwise, Income Level = Low

```
Customer Income = SWITCH(True(), [AnnualIncome] >=150000, "Very High", [AnnualIncome] >=100000, "High", [AnnualIncome] >=50000, "Average", "Low" )
```

AnnualIncome	TotalChildren	EducationLevel	Occupation	HomeOwner	Domain Name	Customer Priority	Customer Income
20000	1	Bachelors	Manual	Y	Adventure Works	Standard	Low
10000	1	Bachelors	Manual	Y	Adventure Works	Standard	Low
10000	1	Bachelors	Manual	Y	Adventure Works	Standard	Low
170000	0	Partial College	Management	N	Adventure Works	Standard	Very High
100000	3	Partial College	Management	Y	Adventure Works	Standard	High
100000	3	Partial College	Management	Y	Adventure Works	Standard	High
110000	0	Partial College	Management	N	Adventure Works	Standard	High
110000	0	Partial College	Management	N	Adventure Works	Standard	High
100000	3	Partial College	Management	Y	Adventure Works	Standard	High
100000	3	Partial College	Management	Y	Adventure Works	Standard	High

- Use a SWITCH function* to create another column named Education Category:
  → If EducationLevel is High School or Partial High School, Education Category = High School
  → If EducationLevel is Bachelors or Partial College, Education Category = Undergrad

→ If EducationLevel is Graduate Degree, Education Category = Graduate

```
Education Category = SWITCH( True(), [EducationLevel]="High School" || [EducationLevel]= "Partial High School", "High School",
[EducationLevel]= "Bachelors" || [EducationLevel]= "Partial College", "Undergrad", [EducationLevel]="Graduate Degree", "Graduate", Blank() )
```

EducationLevel	Occupation	HomeOwner	Domain Name	Customer Priority	Customer Income	Education Category
Partial College	Skilled Manual	Y	Adventure Works	Standard	Average	Undergrad
Partial College	Skilled Manual	N	Adventure Works	Standard	Average	Undergrad
Bachelors	Skilled Manual	Y	Adventure Works	Standard	Average	Undergrad
Graduate Degree	Skilled Manual	Y	Adventure Works	Standard	Average	Graduate
Graduate Degree	Skilled Manual	Y	Adventure Works	Standard	Average	Graduate
Partial College	Skilled Manual	Y	Adventure Works	Standard	Average	Undergrad
Partial College	Skilled Manual	Y	Adventure Works	Standard	Average	Undergrad
High School	Skilled Manual	Y	Adventure Works	Standard	Low	High School
Bachelors	Skilled Manual	Y	Adventure Works	Standard	Average	Undergrad
Partial College	Skilled Manual	Y	Adventure Works	Standard	Average	Undergrad

b) Text Function

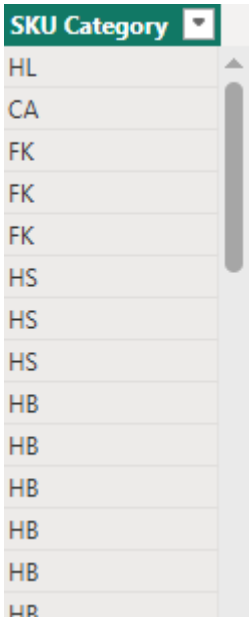
- Update the Month Short column in the Calendar Lookup table to extract and capitalize the first 3 characters of the month name

```
Month Short = UPPER( MID([Date].[Month], 1, 3))
```

Month Short
JAN
JAN
JAN
JAN
JAN
JAN
JAN
FEB
FEB
FEB
FEB
FEB
FEB

- Create a new column in the Product Lookup table named SKU Category, to return any number of characters before the first hyphen in the ProductSKU column

```
SKU Category = LEFT([SKU Type] , FIND( "-", [SKU Type]) -1)
```



SKU Category
HL
CA
FK
FK
FK
HS
HS
HS
HB
HB
HB
HB
HB
HB

c) Date function

- Create a new column in the Customer Lookup table named Birth Year,to extract only the year from the BirthDate column.

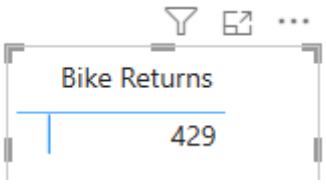
```
Birth Year = YEAR([BirthDate])
```

Birth Year
1964
1964
1963
1974
1965
1963
1975
1976
1976
1974
1963
1963

d) Calculate Function

- Create a new measure named Bike Returns to calculate the total quantity of bikes returned

```
Bike Returns = CALCULATE(SUM('AdventureWorks Returns Data'[ReturnQuantity]),
'AdventureWorks Product Subcategories Lookup'[ProductSubcategoryKey] IN {1,2,3} )
```



- Create a matrix to show Bike Returns (values) by Start of Month (rows). What do you notice about the volume of bike returns over time?

StartOfMonth	Bike Returns
01-01-2020	4
01-01-2021	8
01-01-2022	14
01-02-2020	4
01-02-2021	8
01-02-2022	22
01-03-2020	9
01-03-2021	8
01-03-2022	27
01-04-2020	14
01-04-2021	5
01-04-2022	38
01-05-2020	11
01-05-2021	10
01-05-2022	36
01-06-2020	4
01-06-2021	8
01-06-2022	34
01-07-2020	3
01-07-2021	12
01-08-2020	6
01-08-2021	14
01-09-2020	2
01-09-2021	22
01-10-2020	11
01-10-2021	26
01-11-2020	5
01-11-2021	25
01-12-2020	13
01-12-2021	26
Total	**429**

- Create a new measure named Bike Sales to calculate the total quantity of bikes sold, and add it to the matrix. What do you notice?

```
Bike Sales = CALCULATE( SUM('AdventureWorks Sales Data'[OrderQuantity]),
'AdventureWorks Product Lookup'[ProductSubcategoryKey] IN { 1,2,3})
```

StartOfMonth	Bike Returns	Bike Sales
01-01-2020	4	184
01-01-2021	8	242
01-01-2022	14	766
01-02-2020	4	165
01-02-2021	8	267
01-02-2022	22	806
01-03-2020	9	198
01-03-2021	8	266
01-03-2022	27	888
01-04-2020	14	204
01-04-2021	5	290
01-04-2022	38	956
01-05-2020	11	206
01-05-2021	10	329
01-05-2022	36	1116
01-06-2020	4	212
01-06-2021	8	312
01-06-2022	34	1157
01-07-2020	3	247
01-07-2021	12	506
01-08-2020	6	278
01-08-2021	14	485
01-09-2020	2	196
01-09-2021	22	575
01-10-2020	11	223
01-10-2021	26	612
01-11-2020	5	191
01-11-2021	25	688
01-12-2020	13	326
01-12-2021	26	1038
Total	**429**	**13929**

- Create a new measure named Bike Return Rate using either CALCULATE or DIVIDE, and add it to the matrix

```
Bike Return Rate = DIVIDE('AdventureWorks Returns Data'[Bike Returns], 'AdventureWorks Sales Data'[Bike Sales],0 )
```

StartOfMonth	Bike Returns	Bike Sales	Bike Return Rate
01-01-2020	4	184	0.02
01-01-2021	8	242	0.03
01-01-2022	14	766	0.02
01-02-2020	4	165	0.02
01-02-2021	8	267	0.03
01-02-2022	22	806	0.03
01-03-2020	9	198	0.05
01-03-2021	8	266	0.03
01-03-2022	27	888	0.03
01-04-2020	14	204	0.07
01-04-2021	5	290	0.02
01-04-2022	38	956	0.04
01-05-2020	11	206	0.05
01-05-2021	10	329	0.03
01-05-2022	36	1116	0.03
01-06-2020	4	212	0.02
01-06-2021	8	312	0.03
01-06-2022	34	1157	0.03
01-07-2020	3	247	0.01
01-07-2021	12	506	0.02
01-08-2020	6	278	0.02
01-08-2021	14	485	0.03
01-09-2020	2	196	0.01
01-09-2021	22	575	0.04
01-10-2020	11	223	0.05
01-10-2021	26	612	0.04
01-11-2020	5	191	0.03
01-11-2021	25	688	0.04
01-12-2020	13	326	0.04
01-12-2021	26	1038	0.03
Total	**429**	**13929**	**0.03**

e) Calculate & all function

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

```
All returns = CALCULATE(SUM('AdventureWorks Returns Data'[ReturnQuantity]), ALL('AdventureWorks Returns Data'))
```

All returns
1828

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

```
% All returns = DIVIDE([Total returns], [All returns], 0)
```

% All returns
0.42

- Create a matrix to show % of All Returns (values) by product Category Name (rows).

CategoryName	% All returns
Accessories	0.43
Bikes	0.40
Clothing	0.40
Total	**0.42**

- Which category accounts for the largest percentage of returns? The smallest?

f) Iterator Function

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

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

Total Cost
1,44,56,871.39

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

```
Total Profit = SUMX('AdventureWorks Sales Data', ([OrderQuantity] * RELATED('AdventureWorks Product Lookup'[ProductPrice]))-
([OrderQuantity] * RELATED('AdventureWorks Product Lookup'[ProductCost])))
```

Total Profit
1,04,57,715.43

- Create a matrix to show Total Profit (values) by Year (rows).

Year	Total Profit
2020	26,01,602.33
2021	39,67,084.13
2022	38,89,028.97
Total	**1,04,57,715.43**

- How much profit has AdventureWorks earned so far in 2022

Year	Total Profit
2022	38,89,028.97
Total	**38,89,028.97**

g) Time Intelligence Function

- Add the following measures to the model:
  Previous Month Returns

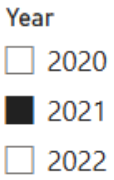
```
Prev Month Returns = CALCULATE(SUM('AdventureWorks Returns Data'[ReturnQuantity]), PREVIOUSMONTH('AdventureWorks Calendar Lookup'[Date]))
```

Month	Prev Month Returns	Total returns
November	137	136
December	136	163
October	122	137
September	120	122
August	45	120
January	13	8
June	10	8
February	8	8
March	8	8
April	8	5
July	8	45
May	5	10
Total	**13**	**770**

Previous Month Orders

```
Prev Month Orders = CALCULATE(SUM([OrderQuantity]), PREVIOUSMONTH('AdventureWorks Calendar Lookup'[Date]))
```

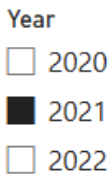
Month	Prev Month Orders	Total orders
January	326	242
February	242	267
March	267	266
April	266	290
May	290	329
June	329	312
July	312	1954
August	1954	5958
September	5958	5970
October	5970	6387
November	6387	6332
December	6332	7923
Total	**326**	**36230**



Previous Month Profit

```
Prev Month Profit = CALCULATE([Total Profit], PREVIOUSMONTH('AdventureWorks Calendar Lookup'[Date]))
```

Month	Prev Month Profit	Total Profit
January	2,36,830.33	1,82,044.38
February	1,82,044.38	2,00,044.37
March	2,00,044.37	1,99,611.04
April	1,99,611.04	2,09,521.70
May	2,09,521.70	2,33,013.08
June	2,33,013.08	2,27,745.04
July	2,27,745.04	3,42,624.13
August	3,42,624.13	3,48,095.72
September	3,48,095.72	4,10,592.05
October	4,10,592.05	4,41,168.03
November	4,41,168.03	4,82,940.37
December	4,82,940.37	6,89,684.23
Total	**2,36,830.33**	**39,67,084.13**



Order Target (10% increase over previous month)

```
order target = CALCULATE(sum([OrderQuantity])*1.10,DATEADD('AdventureWorks Calendar Lookup'[Date], -1, MONTH))
```

Year	Sum of OrderQuantity	order target
2020	**2630**	**2,534.40**
January	184	
February	165	202.40
March	198	181.50
April	204	217.80
May	206	224.40
June	212	226.60
July	247	233.20
August	278	271.70
September	196	305.80
October	223	215.60
November	191	245.30
December	326	210.10
2021	**36230**	**31,496.30**
January	242	358.60
February	267	266.20
March	266	293.70
April	290	292.60
May	329	319.00
June	312	361.90
July	1954	343.20
August	5958	2,149.40
September	5970	6,553.80
October	6387	6,567.00
November	6332	7,025.70
December	7923	6,965.20
2022	**45314**	**49,474.70**
January	7020	8,715.30
February	6828	7,722.00
March	7327	7,510.80
April	7680	8,059.70
May	8199	8,448.00
June	8260	9,018.90
Total	**84174**	**83,505.40**

Profit Target (10% increase over previous month)

```
profit target = CALCULATE([Total Profit]*1.10, DATEADD('AdventureWorks Calendar Lookup'[Date], -1, MONTH))
```

Year	Total Profit	profit target
2020	**26,01,602.33**	**26,01,249.20**
January	2,35,814.03	
February	2,12,186.69	2,59,395.43
March	2,59,084.52	2,33,405.36
April	2,63,031.34	2,84,992.97
May	2,66,275.75	2,89,334.48
June	2,70,067.51	2,92,903.33
July	1,96,682.79	2,97,074.26
August	2,18,355.47	2,16,351.07
September	1,40,516.15	2,40,191.02
October	1,68,581.76	1,54,567.76
November	1,34,175.98	1,85,439.94
December	2,36,830.33	1,47,593.58
2021	**39,67,084.13**	**38,65,653.25**
January	1,82,044.38	2,60,513.36
February	2,00,044.37	2,00,248.82
March	1,99,611.04	2,20,048.81
April	2,09,521.70	2,19,572.15
May	2,33,013.08	2,30,473.87
June	2,27,745.04	2,56,314.38
July	3,42,624.13	2,50,519.54
August	3,48,095.72	3,76,886.54
September	4,10,592.05	3,82,905.29
Total	**1,04,57,715.43**	**1,06,54,751.96**

90-day Rolling Profit

```
90-days rolling profit = CALCULATE([Total Profit],
DATESINPERIOD('AdventureWorks Sales Data'[OrderDate],MAX('AdventureWorks Sales Data'[OrderDate]),-90,DAY))
```

OrderDate	Total Profit	90-days rolling profit
01 January 2020	3,455.78	3,455.78
02 January 2020	5,627.90	9,083.68
03 January 2020	11,476.51	20,560.18
04 January 2020	7,115.74	27,675.92
05 January 2020	3,099.90	30,775.83
06 January 2020	8,511.77	39,287.60
07 January 2020	3,385.86	42,673.46
08 January 2020	10,285.56	52,959.02
09 January 2020	5,627.90	58,586.92
10 January 2020	5,697.82	64,284.74
11 January 2020	12,883.48	77,168.23
12 January 2020	9,848.83	87,017.06
13 January 2020	3,099.90	90,116.96
14 January 2020	12,905.36	1,03,022.32
15 January 2020	9,018.43	1,12,040.75
16 January 2020	9,999.61	1,22,040.36
17 January 2020	10,285.56	1,32,325.92
18 January 2020	6,199.81	1,38,525.73
19 January 2020	7,320.83	1,45,846.56
20 January 2020	8,441.85	1,54,288.41
21 January 2020	10,010.55	1,64,298.96
22 January 2020	12,894.42	1,77,193.38
23 January 2020	4,220.93	1,81,414.31
24 January 2020	10,215.64	1,91,629.95
25 January 2020	2,264.83	1,93,894.79
26 January 2020	5,778.68	1,99,673.47
27 January 2020	5,994.71	2,05,668.18
28 January 2020	5,983.77	2,11,651.96
29 January 2020	4,587.74	2,16,239.70
30 January 2020	10,706.69	2,26,946.38
Total	**1,04,57,715.43**	**21,42,663.11**