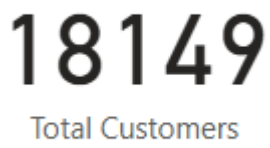
DAX Assignment

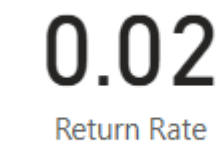
- Create a measure named Total Customers, to calculate the number of distinct AdventureWorks customers who made a transaction.

```
Total Customers = DISTINCTCOUNT('AdventureWorks Customer Lookup'[CustomerKey])
```



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

```
Return Rate = 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(
    'AdventureWorks Customer Lookup'[TotalChildren] > 0 && 'AdventureWorks Customer Lookup'[AnnualIncome] > 100000, "Priority",
    "Standard"
    )
```

Customer Priority
Standard
Standard
Standard
Standard
Standard
Priority
Priority
Standard
Standard
Standard
Standard
Standard
Standard
Priority
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(),
                    'AdventureWorks Customer Lookup'[AnnualIncome] >= 150000, "Very High",
                    'AdventureWorks Customer Lookup'[AnnualIncome] >= 100000, "High",
                    'AdventureWorks Customer Lookup'[AnnualIncome] >= 50000, "Average",
                    "Low"
                     )
```

Customer Income
High
Low
Average
Average
Very High
Very High
Low
Low
Low
Low
Low
Average
Average
Average
Average
Average
Average
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(),
                        'AdventureWorks Customer Lookup'[EducationLevel] = "High School" || 'AdventureWorks Customer Lookup'[EducationLevel] = "Partial High School", "High School",
                        'AdventureWorks Customer Lookup'[EducationLevel] = "Bachelors" || 'AdventureWorks Customer Lookup'[EducationLevel] = "Partial College", "Undergrad",
                        "Graduate"
                    )
```

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(LEFT(FORMAT('AdventureWorks Calendar Lookup'[Date], "MMMM"), 3))
```

Date	Month Short
01 January 2020	JAN
02 January 2020	JAN
03 January 2020	JAN
04 January 2020	JAN
05 January 2020	JAN
06 January 2020	JAN
07 January 2020	JAN
08 January 2020	JAN
09 January 2020	JAN
10 January 2020	JAN
11 January 2020	JAN
12 January 2020	JAN
13 January 2020	JAN
14 January 2020	JAN
15 January 2020	JAN

- 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(
    'AdventureWorks Product Lookup'[ProductSKU],
    FIND("-", 'AdventureWorks Product Lookup'[ProductSKU]) - 1
)
```

SKU Category
HL
HL
HL
CA
FK
FK
FK
HS
HS

c) Date Function

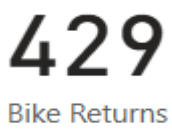
- Create a new column in the Customer Lookup table named Birth Year, to extract only the year from the BirthDate column.

```
Birth Year = YEAR('AdventureWorks Customer Lookup'[BirthDate])
```

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 Categories Lookup'[CategoryName] = "Bikes")
```



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

Year	Bike Returns
2020	86
January	4
1	4
February	4
1	4
March	9
1	9
April	14
1	14
May	11
1	11
June	4
1	4
July	3
1	3
August	6
1	6
September	2
1	2
October	11
1	11
November	5
1	5
December	13
1	13
2021	172
January	8
1	8
February	8
1	8
March	8
1	8
April	5
1	5
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 Categories Lookup'[CategoryName] = "Bikes")
```

Year	Bike Sales
2020	**2630**
January	184
February	165
March	198
April	204
May	206
June	212
July	247
August	278
September	196
October	223
November	191
December	326
2021	**5610**
January	242
February	267
March	266
April	290
May	329
June	312
July	506
August	485
September	575
October	612
November	688
December	1038
2022	**5689**
Total	**13929**

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

```
Bike Return Rate = [Bike Returns] / [Bike Sales]
```

Year	Bike Return Rate
2020	**3.27%**
January	2.17%
February	2.42%
March	4.55%
April	6.86%
May	5.34%
June	1.89%
July	1.21%
August	2.16%
September	1.02%
October	4.93%
November	2.62%
December	3.99%
2021	**3.07%**
January	3.31%
February	3.00%
March	3.01%
April	1.72%
May	3.04%
June	2.56%
July	2.37%
August	2.89%
September	3.83%
October	4.25%
November	3.63%
December	2.50%
2022	**3.01%**
Total	**3.08%**

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'))
```

1828

All Returns

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

```
% of All Returns = [All Returns] / SUM('AdventureWorks Sales Data'[OrderQuantity])
```

2.17%

% of All Returns

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

CategoryName	% of All Returns
Accessories	3.16%
Bikes	13.12%
Clothing	14.70%
Components	Infinity
Total	**2.17%**

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

14.46M

Total Cost

- 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])))
```

- 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

2022	38,89,028.97

g) Time Intelligence Function

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

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

Month	Total Returns	Previous Month Returns
January	158	163
February	154	158
March	160	154
April	164	160
May	169	164
June	167	169
Total	**972**	**163**

- Previous Month Returns

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

Month	Sum of OrderQuantity	Previous Month Orders
January	7020	7923
February	6828	7020
March	7327	6828
April	7680	7327
May	8199	7680
June	8260	8199
Total	**45314**	**7923**

- Previous Month Profit

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

Month	Total Profit	Previous Month Profit
January	5,41,843.78	6,89,684.23
February	5,67,573.89	5,41,843.78
March	6,13,453.12	5,67,573.89
April	6,43,826.56	6,13,453.12
May	7,50,754.34	6,43,826.56
June	7,71,577.29	7,50,754.34
Total	**38,89,028.97**	**6,89,684.23**

- Order Target (10% increase over previous month)

```
Order Target = (CALCULATE(SUM('AdventureWorks Sales Data'[OrderQuantity]), PREVIOUSMONTH('AdventureWorks Calendar Lookup'[Date]))) * 1.1
```

Year	Month	Sum of OrderQuantity	Order Target
2020	January	184	
2020	February	165	202.40
2020	March	198	181.50
2020	April	204	217.80
2020	May	206	224.40
2020	June	212	226.60
2020	July	247	233.20
2020	August	278	271.70
2020	September	196	305.80
2020	October	223	215.60
2020	November	191	245.30
2020	December	326	210.10
2021	January	242	358.60
2021	February	267	266.20
2021	March	266	293.70
2021	April	290	292.60
2021	May	329	319.00
2021	June	312	361.90
2021	July	1954	343.20
Total		**84174**	

- Profit Target (10% increase over previous month)

```
Profit Target = (CALCULATE([Total Profit], PREVIOUSMONTH('AdventureWorks Calendar Lookup'[Date]))) * 1.1
```

Year	Month	Total Profit	Profit Target
2020	January	2,35,814.03	
2020	February	2,12,186.69	2,59,395.43
2020	March	2,59,084.52	2,33,405.36
2020	April	2,63,031.34	2,84,992.97
2020	May	2,66,275.75	2,89,334.48
2020	June	2,70,067.51	2,92,903.33
2020	July	1,96,682.79	2,97,074.26
2020	August	2,18,355.47	2,16,351.07
2020	September	1,40,516.15	2,40,191.02
2020	October	1,68,581.76	1,54,567.76
2020	November	1,34,175.98	1,85,439.94
2020	December	2,36,830.33	1,47,593.58
2021	January	1,82,044.38	2,60,513.36
2021	February	2,00,044.37	2,00,248.82
2021	March	1,99,611.04	2,20,048.81
2021	April	2,09,521.70	2,19,572.15
2021	May	2,33,013.08	2,30,473.87
2021	June	2,27,745.04	2,56,314.38
2021	July	3,42,624.13	2,50,519.54
Total		**1,04,57,715.43**	

- 90-day Rolling Profit

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

OrderDate	Total Profit	90-day 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**