DAX Question Sheet

1) In the **DATA** view, create the following **calculated columns**:

- In the **AW_Customer_Lookup** table, add a new column named "**Customer Priority**" that equals "*Priority*" for customers who are under 50 years old and have an annual income of greater than $100,000, and "*Standard*" otherwise

Customer Priority = IF(AW_Customer_Lookup[Current Age] < 50 && AW_Customer_Lookup[AnnualIncome]>100000, "Priority", "Standard")

- In the **AW_Product_Lookup** table, add a new column named "**Price Point**", based on the following criteria
  - *If the product price is greater than $500,* ***Price Point*** *= "High"*
  - *If the product price is between $100 and $500,* ***Price Point*** *= "Mid-Range"*
  - *If the product price is less than or equal to $100,* ***Price Point*** *= "Low"*

Price Point = IF(AW_Product_Lookup[ProductPrice]>500, "High", IF(AW_Product_Lookup[ProductPrice]>100, "Mid-Range", "Low"))

- In the **AW_Calendar_Lookup** table, add a new column named "**Short Day**" to extract and capitalize the first three letters from the **Day Name** column

Short Day = upper(LEFT(AW_Calendar_Lookup[Day Name],3))

- In the **AW_Product_Lookup** table, add a column named "**SKU Category**" to extract the first two characters from the **ProductSKU** field

SKU Category = LEFT(AW_Product_Lookup[ProductSKU], 2)

- ***BONUS:*** *Modify the* ***SKU Category*** *function to return any number of characters up to the first dash (**Hint:** You may need to "search" long and hard for that dash...)*

SKU Category = LEFT(AW_Product_Lookup[ProductSKU], search("-",AW_Product_Lookup[ProductSKU])-1)

2) In the **REPORT** view, create the following **measures** (*Use a matrix visual to match the "**spot check**" values provided*)

- Create a measure named "**Product Models**" to calculate the number of unique product model names

Product Models = DISTINCTCOUNT(AW_Product_Lookup[ModelName])

- Create a measure named "**ALL Returns**" to calculate the grand total number of returns, regardless of the filter context

All Returns = CALCULATE([Total Returns], ALL(AW_Returns))

- Create a measure to calculate "**% of All Returns**"

% Of All_Returns = [Total Returns]/ [All Returns]

- Create a measure named "**Bike Returns**" to calculate total returns for bikes specifically

```
Bike Returns = CALCULATE([Total Returns],
AW_Product_Categories_Lookup[CategoryName] = "Bikes")
```

- Create a measure named "**Total Cost**", by multiplying order quantities by-product costs at the row-level

```
Total Cost = SUMX(AW_Sales, AW_Sales[OrderQuantity] *
RELATED(AW_Product_Lookup[ProductCost]))
```

- Once you've calculated **Total Cost**, create a new measure for "**Total Profit**", defined as the total revenue minus the total cost

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

- Create a measure to calculate Total Orders for the previous month (named "**Prev Month Orders**")

```
Previous Month Orders = CALCULATE([Total Orders],
DATEADD(AW_Calendar_Lookup[Date], -1, MONTH))
```

- Create a measure named "**Order Target**", calculated as a 10% lift over the previous month

```
Order Target = [Previous Month Orders] * 1.10
```

- *Total Returns for the previous month (named "**Prev Month Returns**")*

```
Prev Month Returns = CALCULATE([Total Returns],
DATEADD(AW_Calendar_Lookup[Date], -1, MONTH))
```

- *90-Day Rolling Profit (named "**90-day Rolling Profit**")*

```
90-Day Rolling Profit = CALCULATE([Total Profit],
DATESINPERIOD(AW_Calendar_Lookup[Date],
MAX(AW_Calendar_Lookup[Date]),-90,day))
```