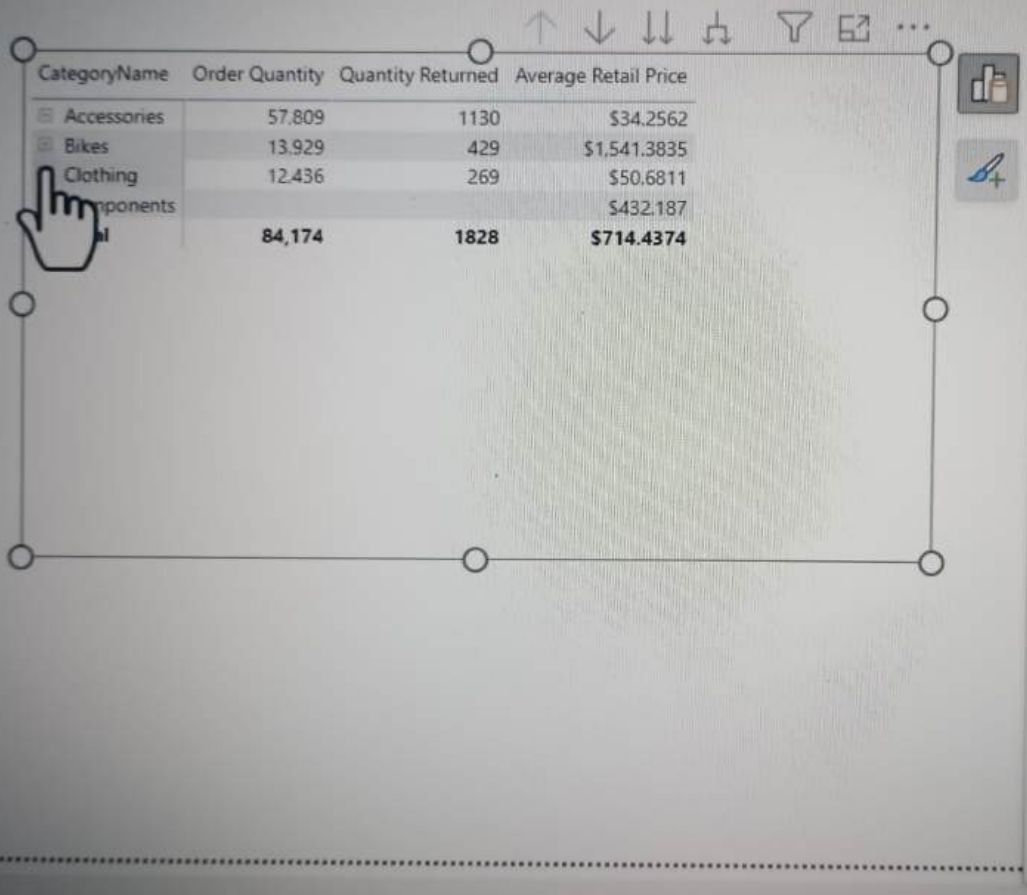

CategoryName
Order Quantity
Quantity Returned
Average Retail Price
Accessories
57,809
1130
$34.2562
Bikes
13,929
429
$1,541.3835
Clothing
12,436
269
$50.6811
$432.187
84,174
1828
$714.4374
Build a visual
Off
Suggest a type
Rows
CategoryName
SubcategoryN...
+Add data
Columns
+Add data
Values
Order Quantity

Home Insert Modeling View Optimize

Total Returns

Format Whole number

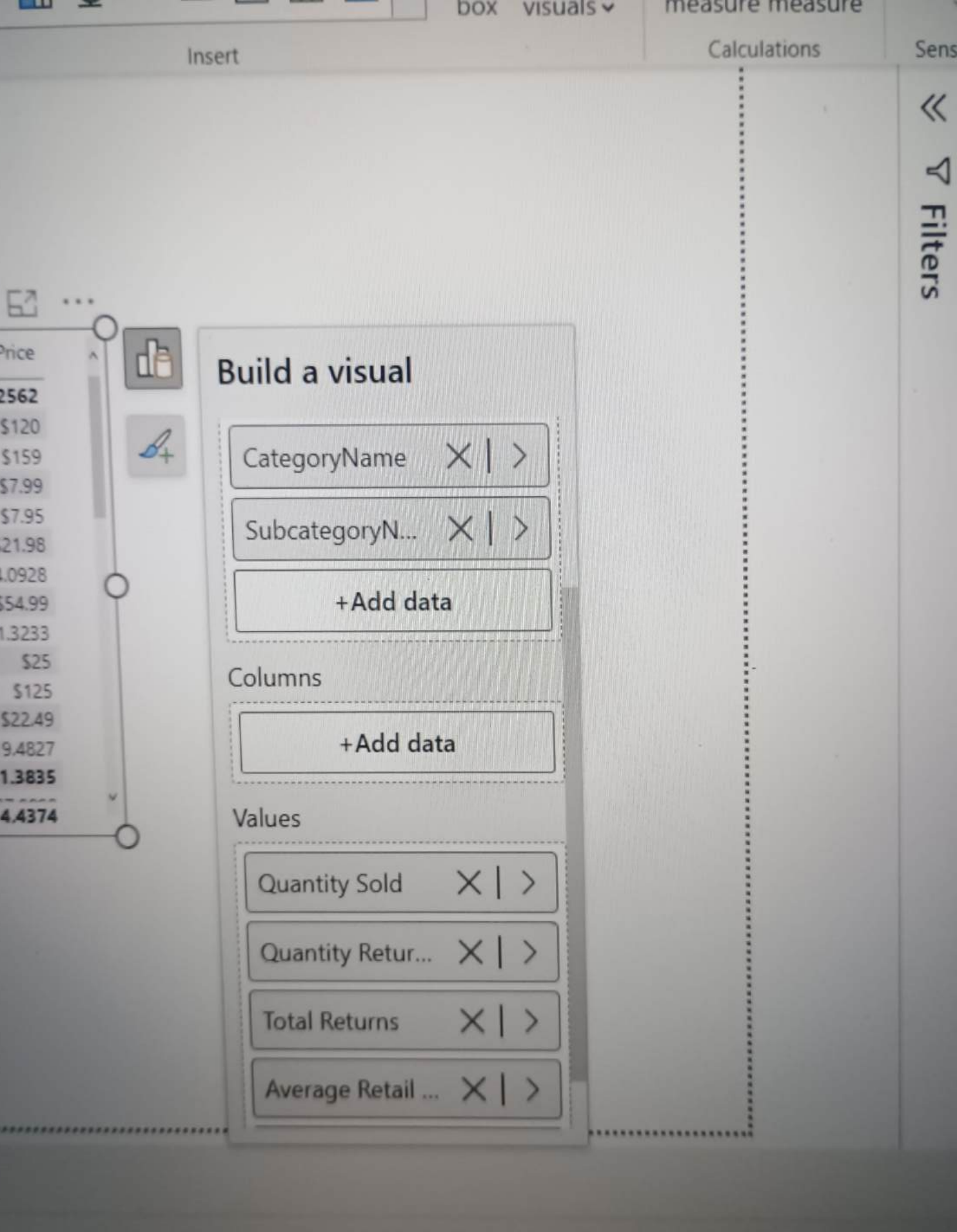
ble Measure Table

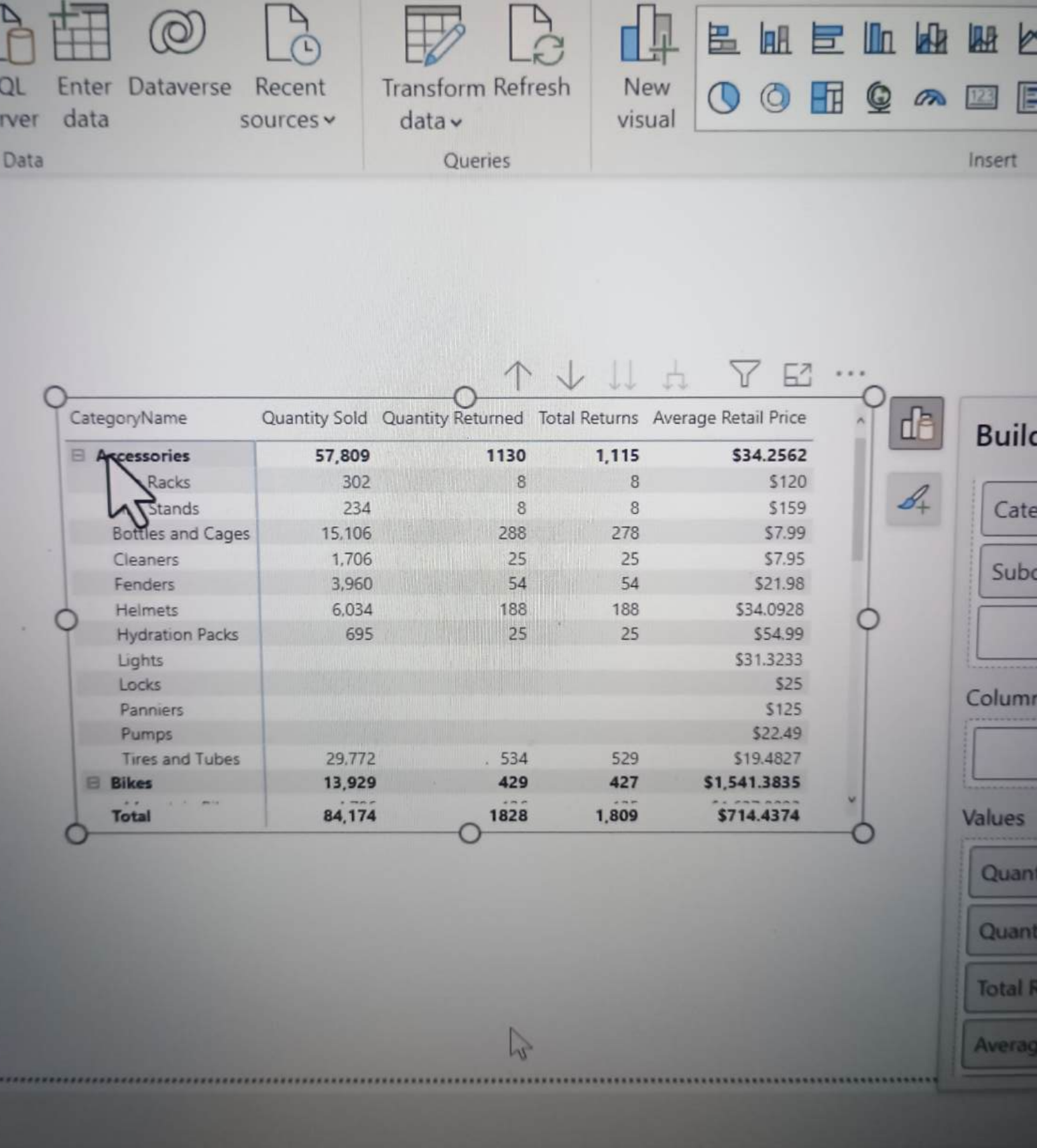
$ % 0

Structure Formatting

```
1 Total Returns =
2 COUNT(
3     'Returns Data'[ReturnQuantity]
4 )
```

CategoryName	Quantity Sold	Quantity
Accessories	**57,809**	
Bike Racks	302	
Bike Stands	234	
Bottles and Cages	15,106	
Cleaners	1,706	
Fenders	3,960	
Helmets	6,034	
Hydration Packs	695	
Lights		
Locks		
Panniers		
Pumps		
Tires and Tubes	29,772	
Bikes	**13,929**	
Total	**84,174**	


box visuals
measure measure
Insert
Calculations
Sens
Filters
Price
2562
$120
$159
$7.99
$7.95
21.98
.0928
$54.99
1.3233
$25
$125
$22.49
9.4827
1.3835
4.4374
Build a visual
CategoryName
SubcategoryN...
+Add data
Columns
+Add data
Values
Quantity Sold
Quantity Retur...
Total Returns
Average Retail ...


QL
rver
Enter data
Dataverse
Recent sources
Data
Transform data
Refresh
Queries
New visual
Insert
CategoryName
Quantity Sold
Quantity Returned
Total Returns
Average Retail Price
Accessories 57,809 1130 1,115 $34.2562
Racks 302 8 8 $120
Stands 234 8 8 $159
Bottles and Cages 15,106 288 278 $7.99
Cleaners 1,706 25 25 $7.95
Fenders 3,960 54 54 $21.98
Helmets 6,034 188 188 $34.0928
Hydration Packs 695 25 25 $54.99
Lights $31.3233
Locks $25
Panniers $125
Pumps $22.49
Tires and Tubes 29,772 534 529 $19.4827
Bikes 13,929 429 427 $1,541.3835
Total 84,174 1828 1,809 $714.4374
Build
Cate
Subc
Column
Values
Quant
Quant
Total F
Averag

Adventure Works Report - Power BI Desktop

Home Insert Modeling View Optimize He

Quantity Sold

table Measure Table

Format Whole number

0

Structure Formatting

```
1 Quantity Sold =
2 SUM(
3     'Sales Data'[OrderQuantity]
4 )
```

CategoryName	Quantity Sold	Quantity Returned
Accessories	57,809	1130
Bikes	13,929	429
Clothing	12,436	269
Components		
Total	**84,174**	**1828**

ome Insert Modeling View Optim

Total Orders

$% Format Whole number

e Measure Table

$ % , .00 0

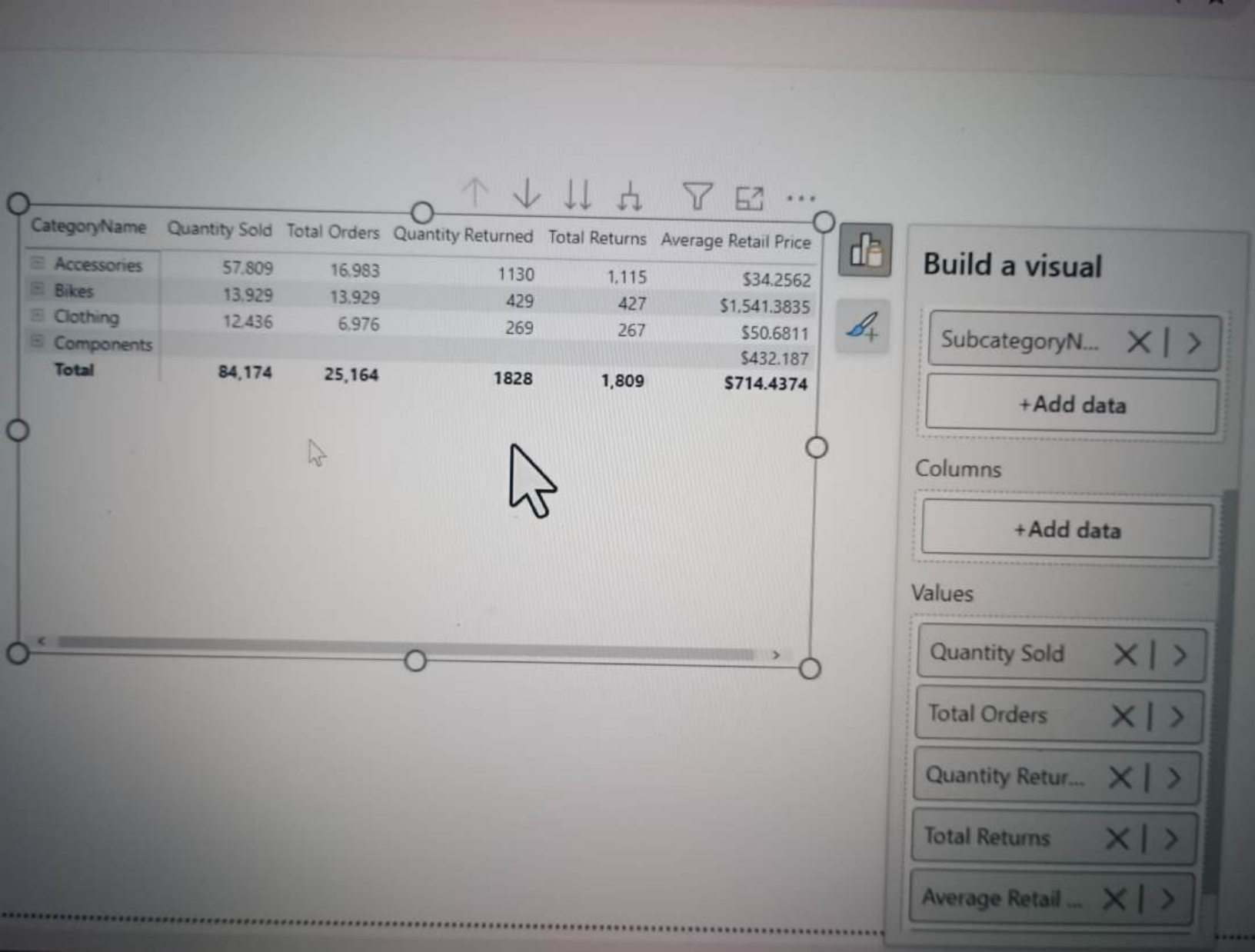
0

Structure

Formatting

```
1 Total Orders =
2 DISTINCTCOUNT(
3     'Sales Data'[OrderNumber]
4 )
```

CategoryName	Quantity Sold	Quar
Accessories	57,809	
Bikes	13,929	
Clothing	12,436	
Components		
Total	**84,174**	



CategoryName	Quantity Sold	Total Orders	Quantity Returned	Total Returns	Average Retail Price
Accessories	57,809	16,983	1130	1,115	$34.2562
Bikes	13,929	13,929	429	427	$1,541.3835
Clothing	12,436	6,976	269	267	$50.6811
Components					$432.187
Total	**84,174**	**25,164**	**1828**	**1,809**	**$714.4374**

Build a visual

SubcategoryN...

+Add data

Columns

+Add data

Values

Quantity Sold

Total Orders

Quantity Retur...

Total Returns

Average Retail ...

NEW MESSAGE

From: **Dianne A. Xu** *(Senior Analyst)*

Subject: **Help with a few measures**

Hey there, excited to start working with you!

I'll need to pull some high-level metrics from our model to share with leadership, and I could use some help with the calculations.

For now, could you please create one measure to calculate the total number of distinct customers, and a second measure that we can use to calculate return rate (quantity returned / quantity sold)? Thank you!

-Dianne

Reply Forward

1.

2.

Key Objectives

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

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

Solution Preview

```
1 Total Customers =
2 DISTINCTCOUNT(
3     'Sales Data'[Customer Key]
4 )
```

```
1 Return Rate =
2 DIVIDE(
3     [Quantity Returned],
4     [Quantity Sold],
5     "No Sales"
6 )
```

Home table | Measure Table

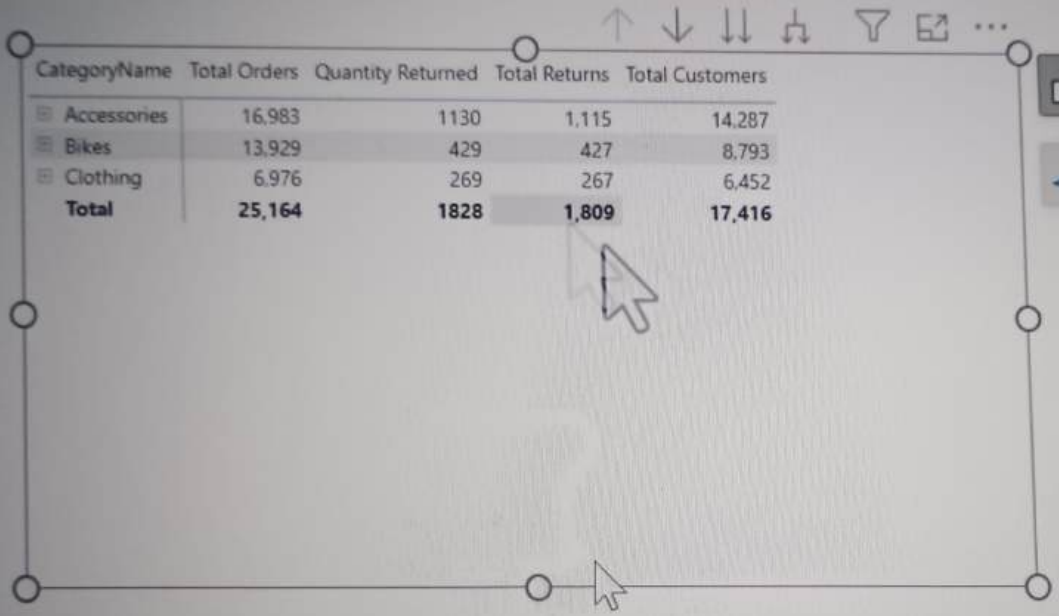
$ % , .00 →.0 | 0

Structure

Formatting

```
1 Total Customers =
2 DISTINCTCOUNT(
3     'Sales Data'[CustomerKey]
4 )
```

CategoryName	Quantity Sold	Total Orders	Q
Accessories	57,809	16,983	
Bikes	13,929	13,929	
Clothing	12,436	6,976	
Components			
Total	**84,174**	**25,164**	



CategoryName	Total Orders	Quantity Returned	Total Returns	Total Customers
Accessories	16,983	1130	1,115	14,287
Bikes	13,929	429	427	8,793
Clothing	6,976	269	267	6,452
Total	**25,164**	**1828**	**1,809**	**17,416**

Build a visual

CategoryName

SubcategoryN...

+Add data

Columns

+Add data

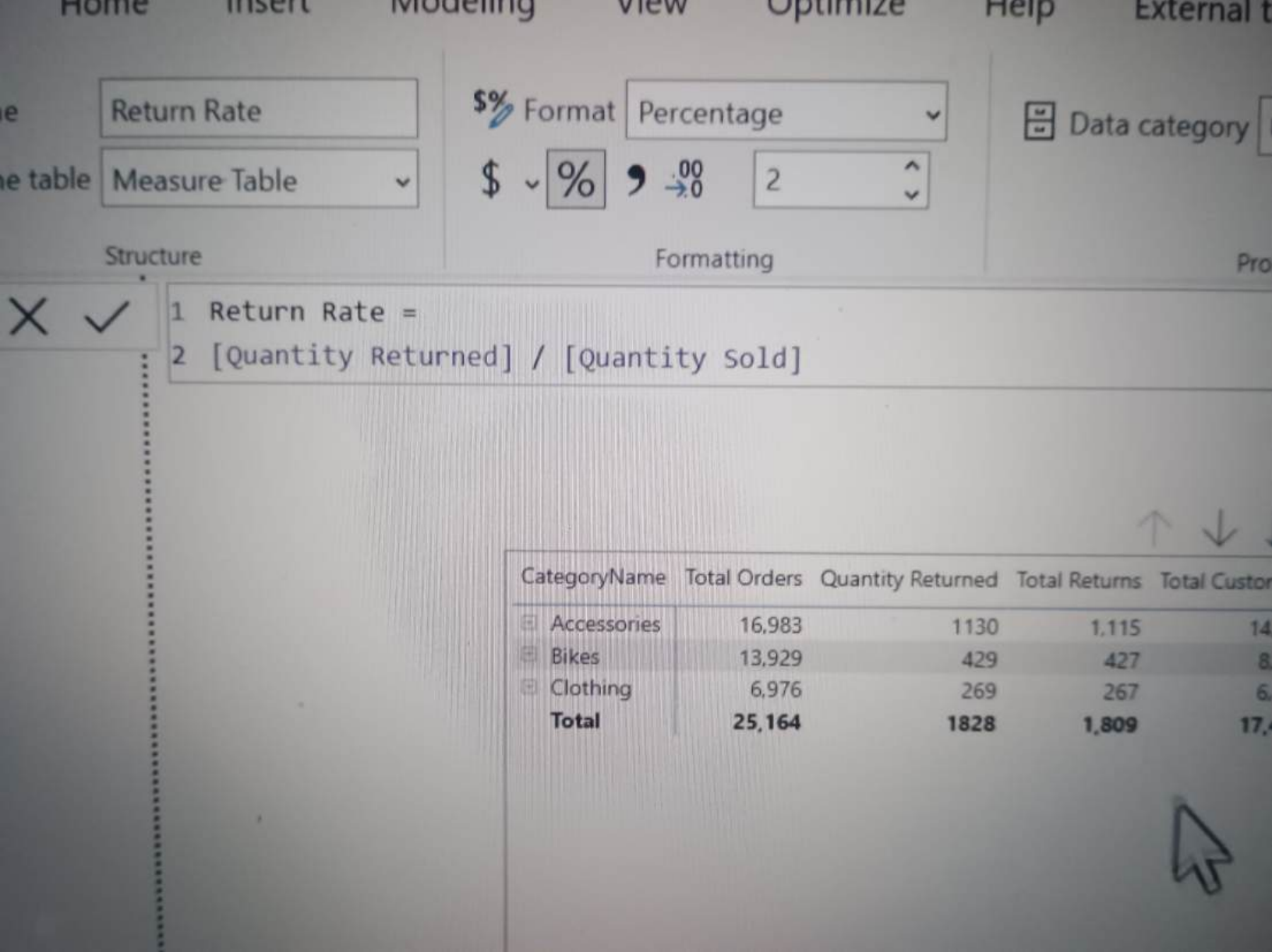
Values

Total Orders

Quantity Retur...

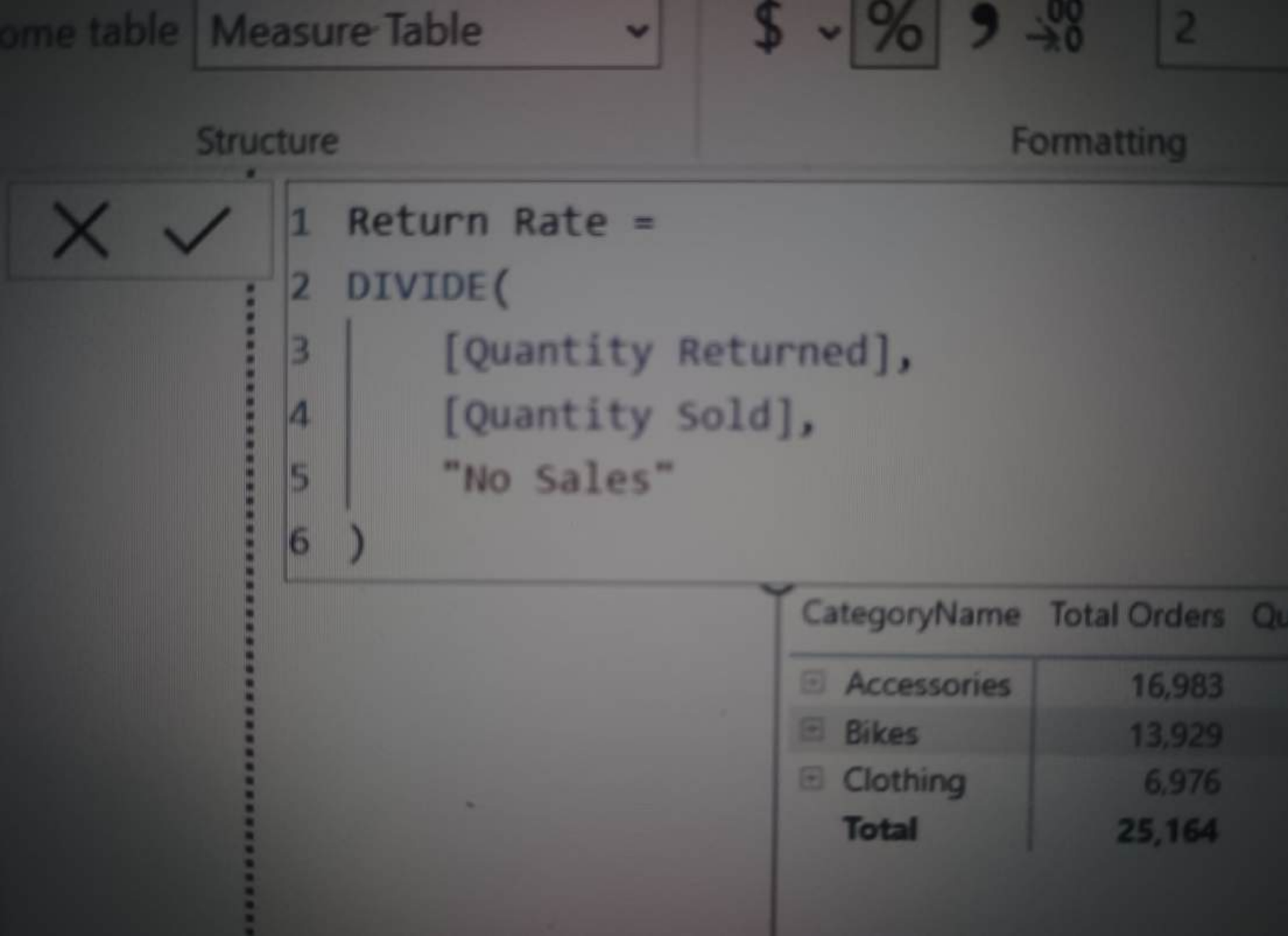
Total Returns

Total Customers


Home
Insert
Modeling
View
Optimize
Help
External t
Return Rate
Format
Percentage
Data category
Measure Table
2
Structure
Formatting
1 Return Rate =
2 [Quantity Returned] / [Quantity Sold]
CategoryName
Total Orders
Quantity Returned
Total Returns
Accessories
16,983
1130
1,115
Bikes
13,929
429
427
Clothing
6,976
269
267
Total
25,164
1828
1,809

urned] / [Quantity Sold]

CategoryName	Total Orders	Quantity Returned	Total Returns	Total Customers	Return Rate
Accessories	16,983	1130	1,115	14,287	1.95%
Bikes	13,929	429	427	8,793	3.08%
Clothing	6,976	269	267	6,452	2.16%
Total	**25,164**	**1828**	**1,809**	**17,416**	**2.17%**


ome table
Measure Table
$
%
2
Structure
Formatting
Return Rate =
DIVIDE(
[Quantity Returned],
[Quantity Sold],
"No Sales"
)
CategoryName
Total Orders
Accessories
16,983
Bikes
13,929
Clothing
6,976
Total
25,164

d],

CategoryName	Total Orders	Quantity Returned	Total Returns	Total Customers	Return Rate
Accessories	16,983	1130	1,115	14,287	1.95%
Bikes	13,929	429	427	8,793	3.08%
Clothing	6,976	269	267	6,452	2.16%
Total	**25,164**	**1828**	**1,809**	**17,416**	**2.17%**

BASIC LOGICAL FUNCTIONS

IF	Checks if a given condition is met and returns one value if the condition is TRUE, and another if the condition is FALSE	=**IF**(LogicalTest, ResultIfTrue, *[ResultIfFalse]*)
IFERROR	Evaluates an expression and returns a specified value if it returns an error, otherwise returns the expression itself	=**IFERROR**(Value, ValueIfError)
SWITCH	Evaluates an expression against a list of values and returns one of multiple possible expressions	=**SWITCH**(Expression, Value1, Result1, ..., *[Else]*)
AND	Checks whether both arguments are TRUE to return TRUE, otherwise returns FALSE	=**AND**(Logical1, Logical2)
OR	Checks whether any argument is TRUE to return TRUE, otherwise returns FALSE	=**OR**(Logical1, Logical2)

***Note**: Use the **&&** and **||** operators to include more than two conditions*

al2)

l2)

Note: *Use the* **&&** *and* **||** *operators to include more than two conditions*

TotalChildren	EducationLevel	Occupation	HomeOwner	Full Name
0	Bachelors	Professional	N	Mr. Shannon
0	Bachelors	Professional	N	Mr. Jessie Liu
0	Bachelors	Professional	N	Mr. Ruben Ka
0	Bachelors	Professional	N	Mr. Ruben M
0	Bachelors	Professional	N	Mr. Joe Rana
0	Bachelors	Professional	N	Mr. Jarrod Su
0	Bachelors	Professional	N	Mr. Dustin G
0	Bachelors	Professional	N	Mr. Clayton J
0	Bachelors	Professional	N	Mr. Irving Sc
0	Bachelors	Professional	N	Mr. Alan Hua
0	Bachelors	Professional	N	Mr. Brendan
0	Bachelors	Professional	N	Mr. Gregory
0	Bachelors	Professional	N	Mr. Marco Va
0	Bachelors	Professional	N	Mr. Alejandro
0	Bachelors	Professional	N	Mr. Shane Fe
0	Bachelors	Professional	N	Mr. Jay Rama
0	Bachelors	Professional	N	Mr. Damien Y
0	Bachelors	Professional	N	Mr. Roy Mart
0	Bachelors	Professional	N	Mr. Pedro Ra
0	Bachelors	Professional	N	Mr. Eugene L
0	Bachelors	Professional	N	Mr. Johnny A
0	Bachelors	Professional	N	Mr. Edwin Zh

Data

Search

- Measure Table
- Calendar Lookup
- Customer Lookup
- Product Categories Lookup
- Product Lookup
- Product Subcategories Look
- Returns Data
- Sales Data
- Territory Lookup

tructure Formatting

```
1 Parent =
2 IF(
3     'Customer Lookup'[TotalChildren] > 0,
4     "Yes",
5     "No"
6 )
```

	AnnualIncome	TotalChildren	EducationLevel
nture-works.com	70000	0	Bachelors
-works.com	70000	0	Bachelors
e-works.com	70000	0	Bachelors
re-works.com	70000	0	Bachelors
works.com	70000	0	Bachelors
e-works.com	70000	0	Bachelors
re-works.com	70000	0	Bachelors

evel	Occupation	HomeOwner	Full Name	Domain Name	Parent ···
	Professional	N	Mr. Shannon Carlson	Adventure Works	No
	Professional	N	Mr. Jessie Liu	Adventure Works	No
	Professional	N	Mr. Ruben Kapoor	Adventure Works	No
	Professional	N	Mr. Ruben Muñoz	Adventure Works	No
	Professional	N	Mr. Joe Rana	Adventure Works	No
	Professional	N	Mr. Jarrod Suri	Adventure Works	No
	Professional	N	Mr. Dustin Goldstein	Adventure Works	No
	Professional	N	Mr. Clayton Jai	Adventure Works	No
	Professional	N	Mr. Irving Schmidt	Adventure Works	No
	Professional	N	Mr. Alan Huang	Adventure Works	No
	Professional	N	Mr. Brendan Raji	Adventure Works	No
	Professional	N	Mr. Gregory Becker	Adventure Works	No
	Professional	N	Mr. Marco Vance	Adventure Works	No
	Professional	N	Mr. Alejandro Hu	Adventure Works	No
	Professional	N	Mr. Shane Fernandez	Adventure Works	No
	Professional	N	Mr. Jay Raman	Adventure Works	No
	Professional	N	Mr. Damien Ye	Adventure Works	No
	Professional	N	Mr. Roy Martinez	Adventure Works	No

SWITCH

SWITCH Evaluates an expression against a list of values and returns one of multiple possible expressions

=**SWITCH**(Expression, Value1, Result1, …, *[Else]*)

Any **DAX expression** that returns a single scalar value, evaluated multiples times

Examples:

- *Calendar[Month ID]*
- *'Product Lookup'[category]*

List of **values** produced by the expression, each paired with a result to return for rows/cases that match

Examples:

```
=SWITCH( Calendar[Month ID],
  1, "January",
  2, "February"
```

Value returned if the expression doesn't match any value argument

PRO TIP

SWITCH(TRUE) is a common DAX pattern to replace multiple nested IF statements

th Name	Month	Start of Year	Year
	7	1/1/2020	2020
	7	1/1/2020	2020
	7	1/1/2020	2020
	7	1/1/2020	2020
	7	1/1/2020	2020
	7	1/1/2020	2020
	7	1/1/2020	2020
	7	1/1/2020	2020
	7	1/1/2020	2020
	7	1/1/2020	2020
	7	1/1/2020	2020
	7	1/1/2020	2020
	7	1/1/2020	2020
	7	1/1/2020	2020
	7	1/1/2020	2020
	7	1/1/2020	2020
	7	1/1/2020	2020
	7	1/1/2020	2020
	7	1/1/2020	2020
	7	1/1/2020	2020
	7	1/1/2020	2020
	7	1/1/2020	2020

Data

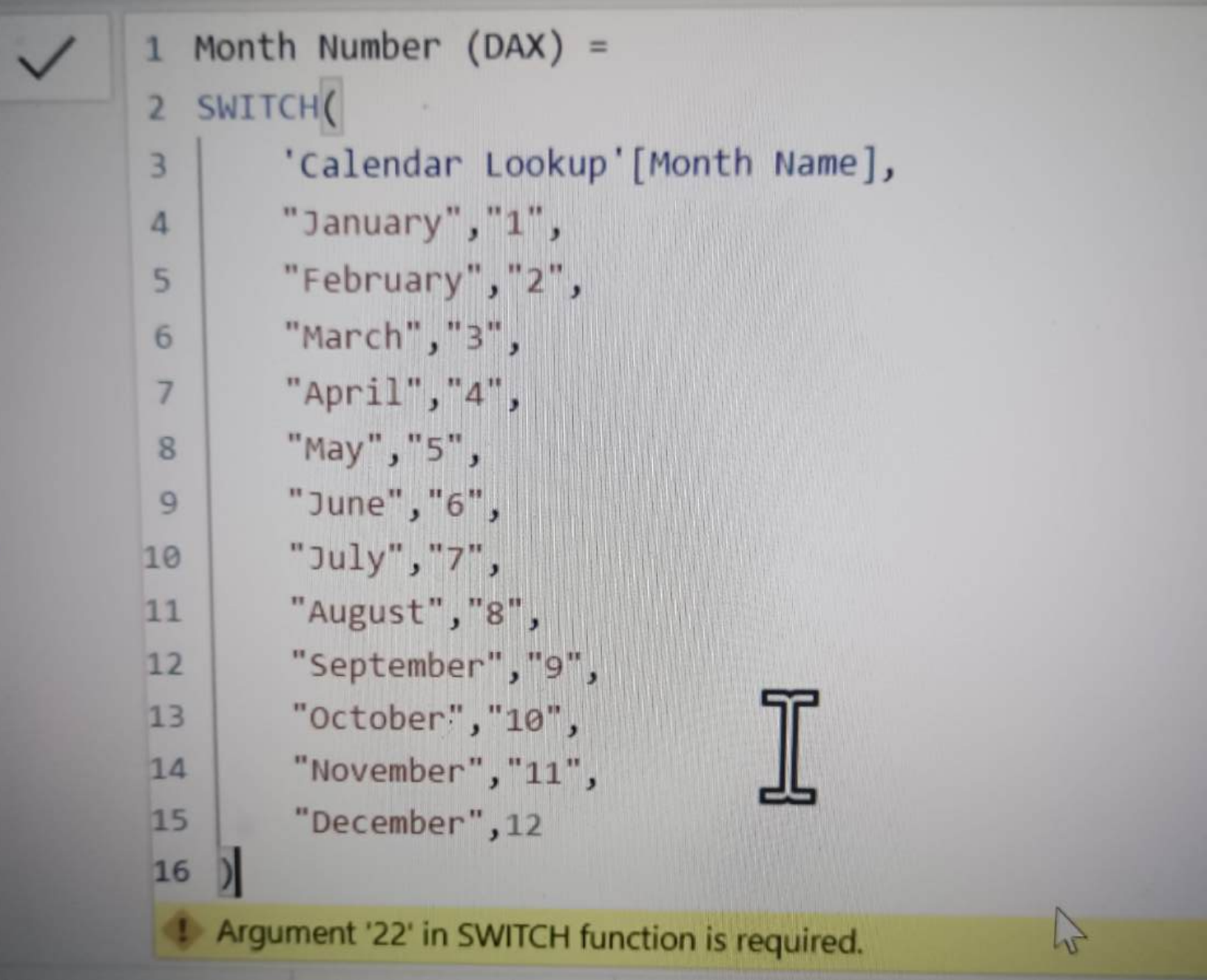
Search

- Measure Table
- Calendar Lookup
- Customer Lookup
- Product Categories Lookup
- Product Lookup
- Product Subcategories Lookup
- Returns Data
- Sales Data
- Territory Lookup

ture | Formatting | Prop

```
1  Month Number (DAX) =
2  IF(
3      'Calendar Lookup'[Month Name] = "January", "1",
4      IF(
5          'Calendar Lookup'[Month Name] = "February", "2",
6          IF(
7              'Calendar Lookup'[Month Name] = "March", "3",
8              "Other"
9          )
10     )
11 )
```

ay Name	Start of Week	Start of Month	Start of Quarter	Month N
ednesday	6/29/2020	7/1/2020	7/1/2020	July
ursday	6/29/2020	7/1/2020	7/1/2020	July
iday	6/29/2020	7/1/2020	7/1/2020	July
turday	6/29/2020	7/1/2020	7/1/2020	July



```
1  Month Number (DAX) =
2  SWITCH(
3      'Calendar Lookup'[Month Name],
4      "January","1",
5      "February","2",
6      "March","3",
7      "April","4",
8      "May","5",
9      "June","6",
10     "July","7",
11     "August","8",
12     "September","9",
13     "October","10",
14     "November","11",
15     "December",12
16 )
```

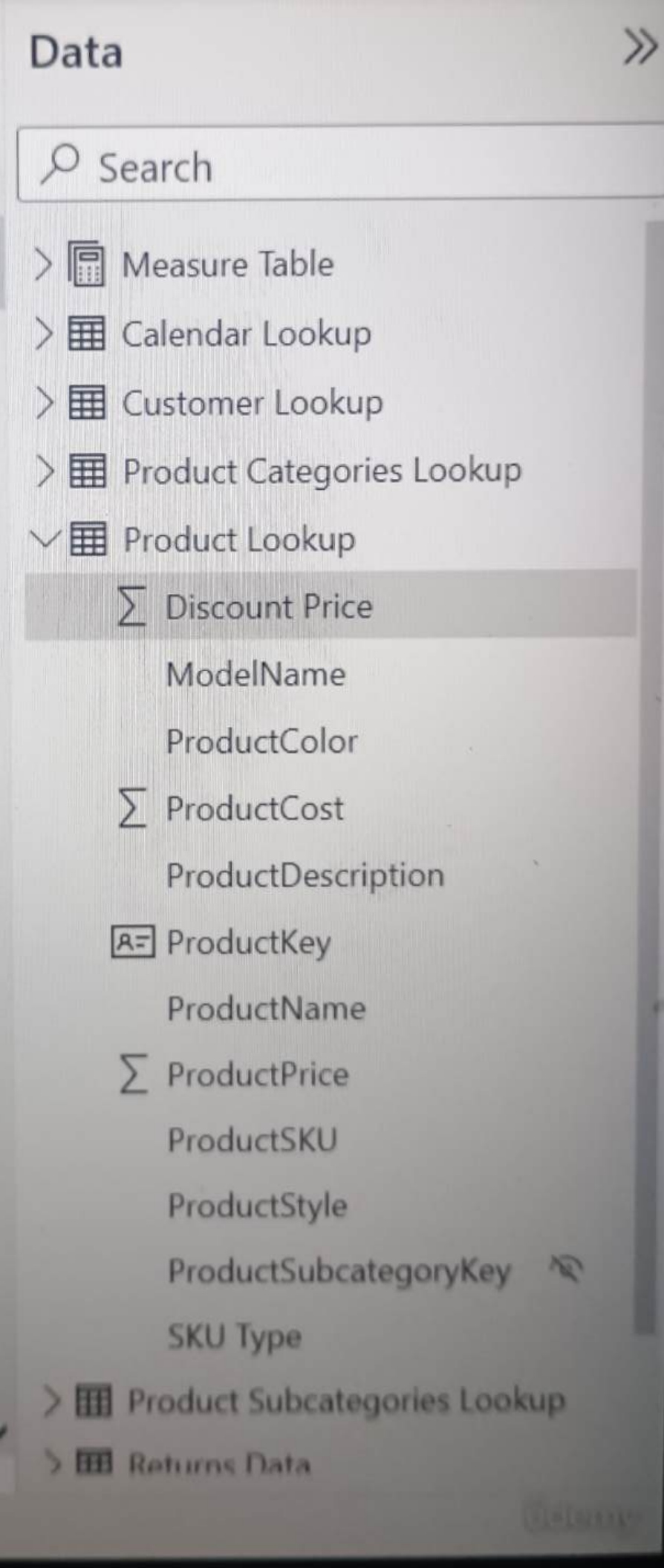
Argument '22' in SWITCH function is required.

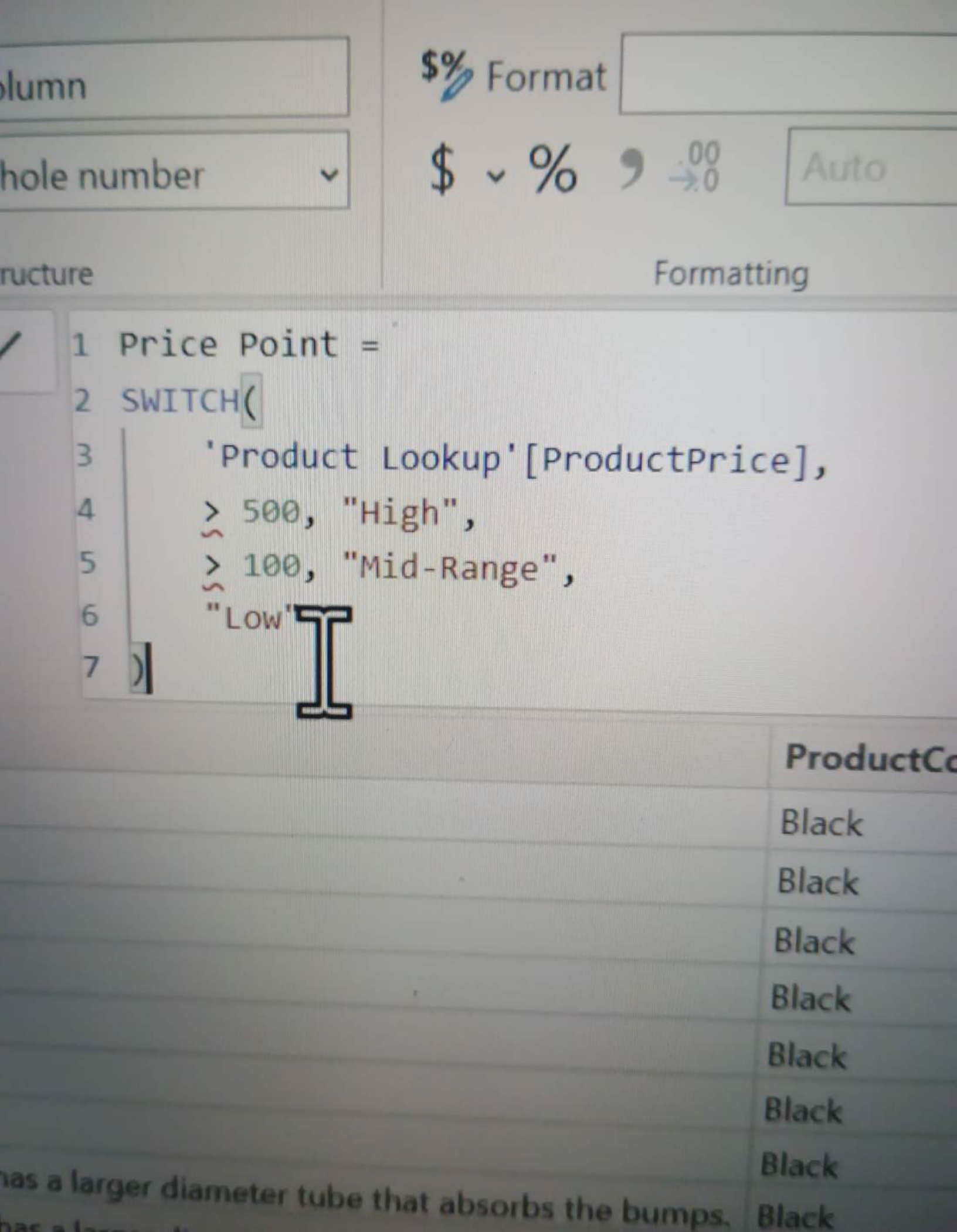
ure

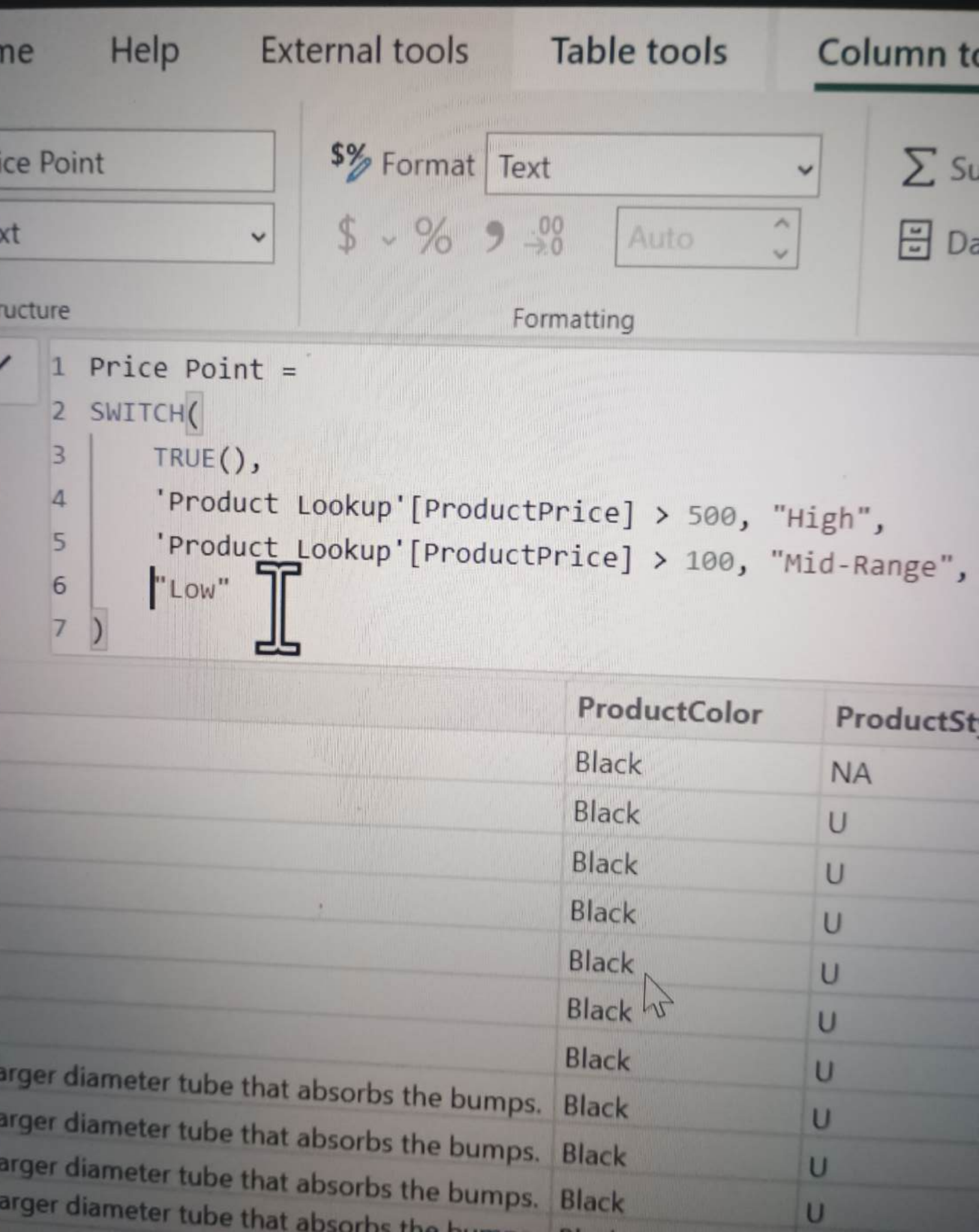
Formatting

```
1  Month Number (DAX) =
2  SWITCH(
3      'Calendar Lookup'[Month Name],
4      "January","1",
5      "February","2",
6      "March","3",
7      "April","4",
8      "May","5",
9      "June","6",
10     "July","7",
11     "August","8",
12     "September","9",
13     "October","10",
14     "November","11",
15     "December","12",
16     "Other"
17 )
```

: Working on it


Data
Search
Measure Table
Calendar Lookup
Customer Lookup
Product Categories Lookup
Product Lookup
Discount Price
ModelName
ProductColor
ProductCost
ProductDescription
ProductKey
ProductName
ProductPrice
ProductSKU
ProductStyle
ProductSubcategoryKey
SKU Type
Product Subcategories Lookup
Returns Data


olumn
$% Format
hole number
$ % , .00 →.0
Auto
ructure
Formatting
1 Price Point =
2 SWITCH(
3 'Product Lookup'[ProductPrice],
4 > 500, "High",
5 > 100, "Mid-Range",
6 "Low"
7)
ProductCo
Black
Black
Black
Black
Black
Black
Black
Black
has a larger diameter tube that absorbs the bumps.


me
Help
External tools
Table tools
Column to
ice Point
xt
Format
Text
Auto
ucture
Formatting
1 Price Point =
2 SWITCH(
3 TRUE(),
4 'Product Lookup'[ProductPrice] > 500, "High",
5 'Product Lookup'[ProductPrice] > 100, "Mid-Range",
6 "Low"
7)
ProductColor
ProductSt
Black NA
Black U
Black U
Black U
Black U
Black U
Black U
arger diameter tube that absorbs the bumps. Black U
arger diameter tube that absorbs the bumps. Black U
arger diameter tube that absorbs the bumps. Black U
arger diameter tube that absorbs the bumps. Black U

NEW MESSAGE

From: **Dianne A. Xu** *(Senior Analyst)*

Subject: **Customer segmentation fields**

Hey there!

Ethan has been working with the DS team on a customer segmentation analysis, and came back to us with a few requests.

Could you please add some new columns in our customer table to identify "priority" customers, segment customers based on income level, and group some of the education categories?

I've attached the logic to use, but reach out with any questions!

-Dianne

Key Objectives

1. 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, Customer Priority = **Priority**
   - Otherwise, Customer Priority = **Standard**
2. Create a calculated column in the Customer Lookup table named **Income Level**:
   - 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**

Key Objectives

BONUS: Use a SWITCH function* to create another column named **Education Category**:

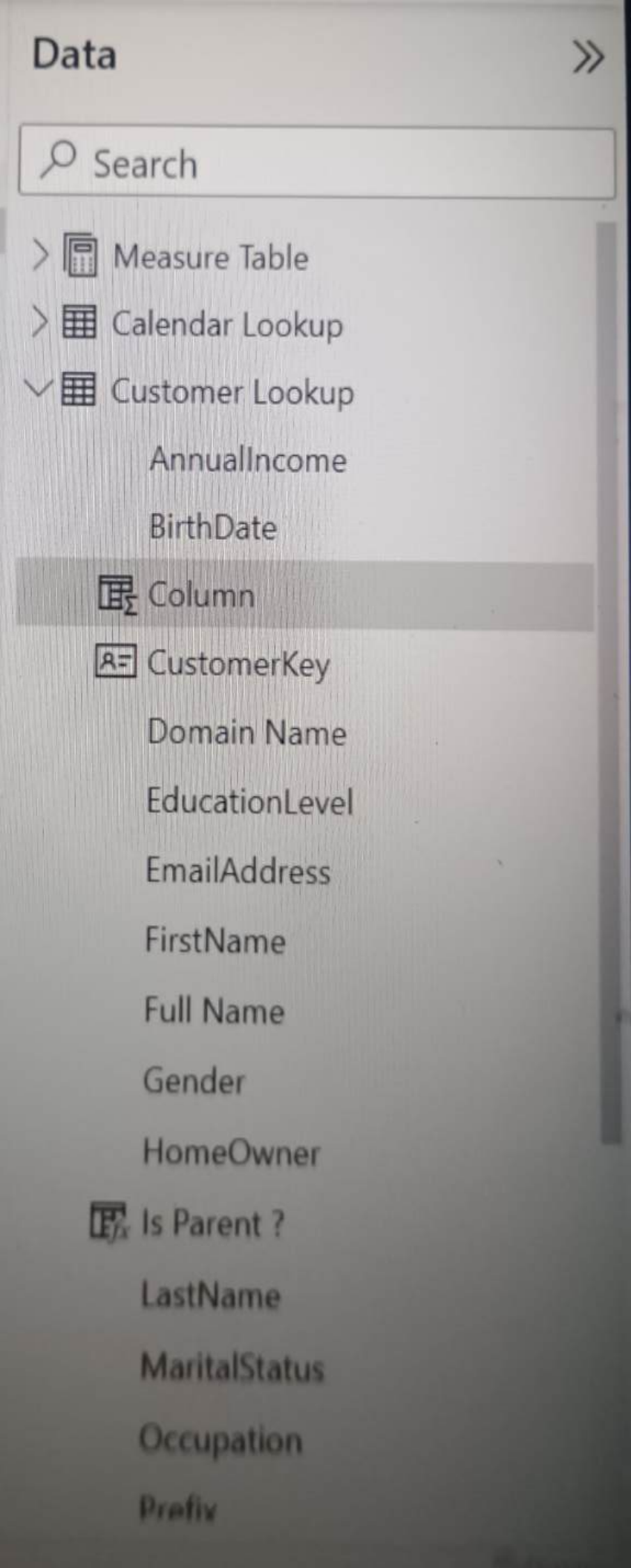
- If EducationLevel is High School or Partial High School, Education Category **= High School**
- If EducationLevel is Bachelors or Partial College, Education Category = **Undergrad**
- If EducationLevel is Graduate Degree, Education Category = **Graduate**

Solution Preview

```
Customer Priority =
IF(
    'Customer Lookup'[AnnualIncome] > 100000 &&
    'Customer Lookup'[Is Parent?] = "Yes",
    "Priority",
    "Standard"
)
```

```
Income Level =
IF('Customer Lookup'[AnnualIncome] >= 150000, "Very High",
IF('Customer Lookup'[AnnualIncome] >= 100000, "High",
IF('Customer Lookup'[AnnualIncome] >= 50000, "Average",
"Low")))
```

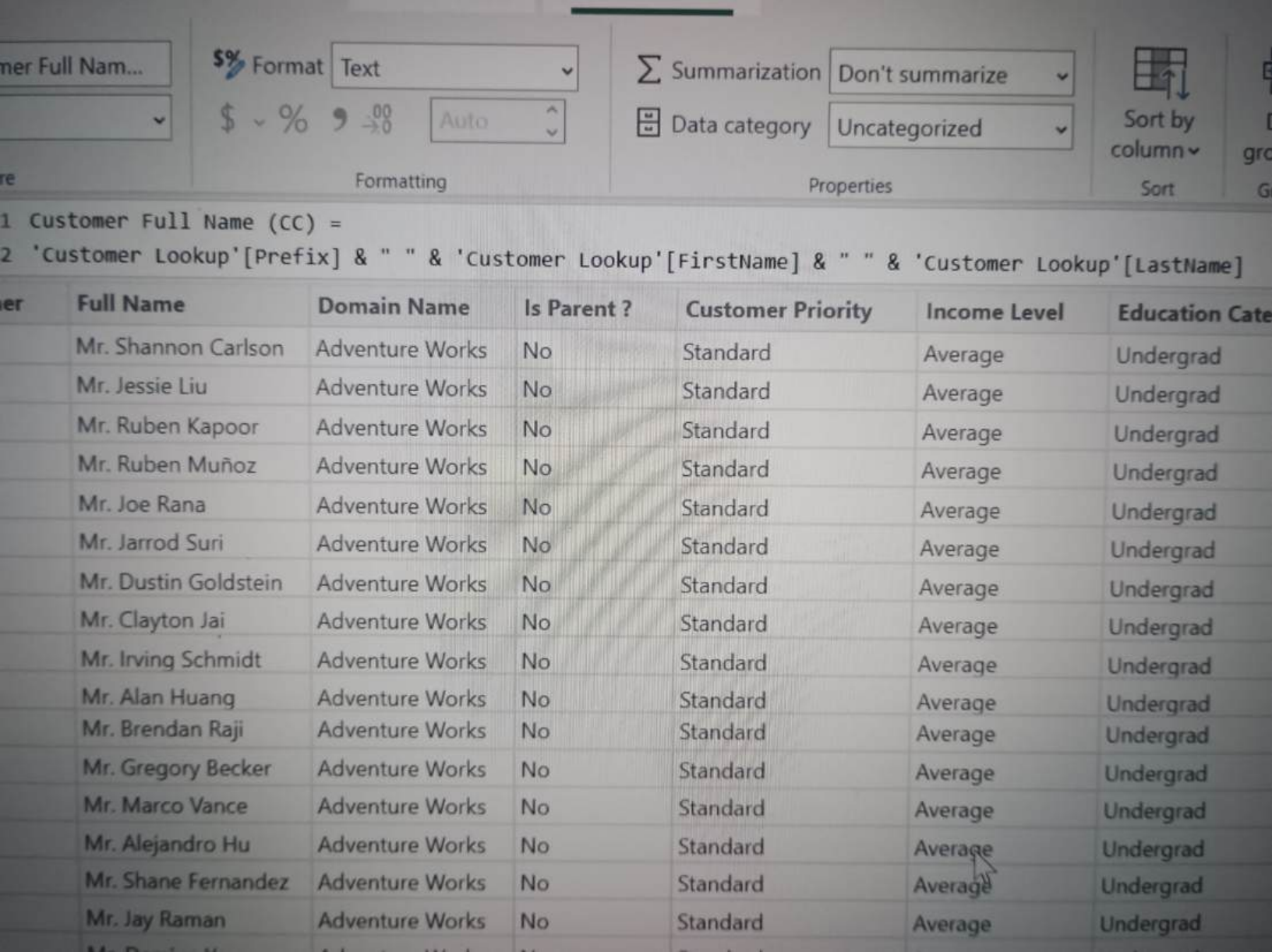
```
Education Category =
SWITCH('Customer Lookup'[EducationLevel],
"High School","High School",
"Partial High School","High School",
"Bachelors","Undergrad",
"Partial College","Undergrad",
"Graduate Degree","Graduate")
```


Data
Search
Measure Table
Calendar Lookup
Customer Lookup
AnnualIncome
BirthDate
Column
CustomerKey
Domain Name
EducationLevel
EmailAddress
FirstName
Full Name
Gender
HomeOwner
Is Parent ?
LastName
MaritalStatus
Occupation
Prefix

TEXT FUNCTIONS

LEN	Returns the number of characters in a string	=**LEN**(Text)
CONCATENATE	Joins two text strings into one	=**CONCATENATE**(Text1, Text2)
UPPER/LOWER	Converts a string to upper or lower case	=**UPPER/LOWER** (Text)
LEFT/RIGHT/MID	Returns a number of characters from the start/middle/end of a text string	=**LEFT/RIGHT**(Text, *[NumChars]*) =**MID**(Text, StartPosition, NumChars)
SUBSTITUTE	Replaces an instance of existing text with new text in a string	=**SUBSTITUTE**(Text, OldText, NewText, *[InstanceNumber]*)
SEARCH	Returns the position where a specified string or character is found, reading left to right	=**SEARCH**(FindText, WithinText, *[StartPosition], [NotFoundValue]*)

Note: *Use the* ***&*** *operator as a shortcut, or to combine more than two strings*



mer Full Nam...

Format Text

Summarization Don't summarize

Data category Uncategorized

Sort by column

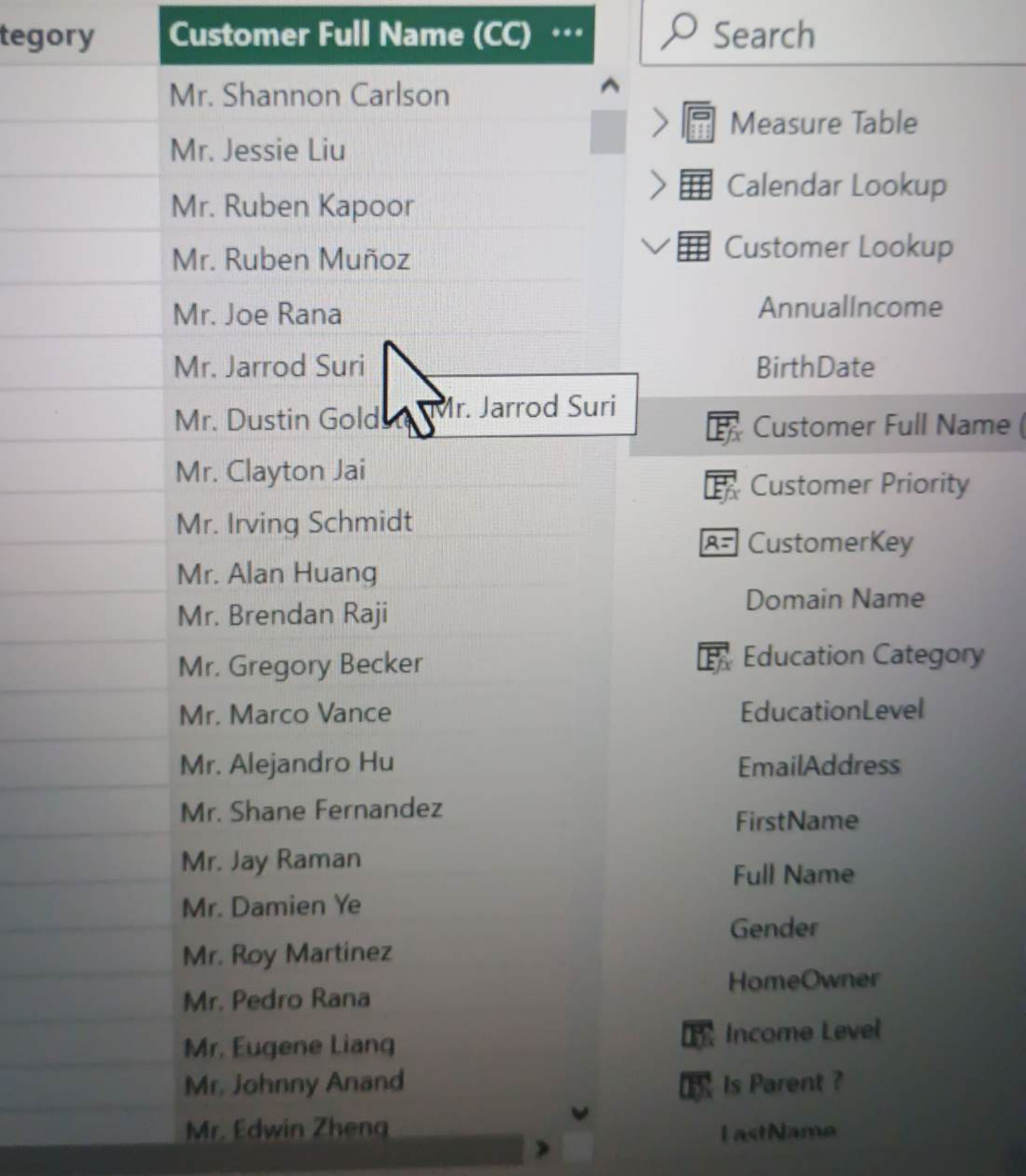
Formatting

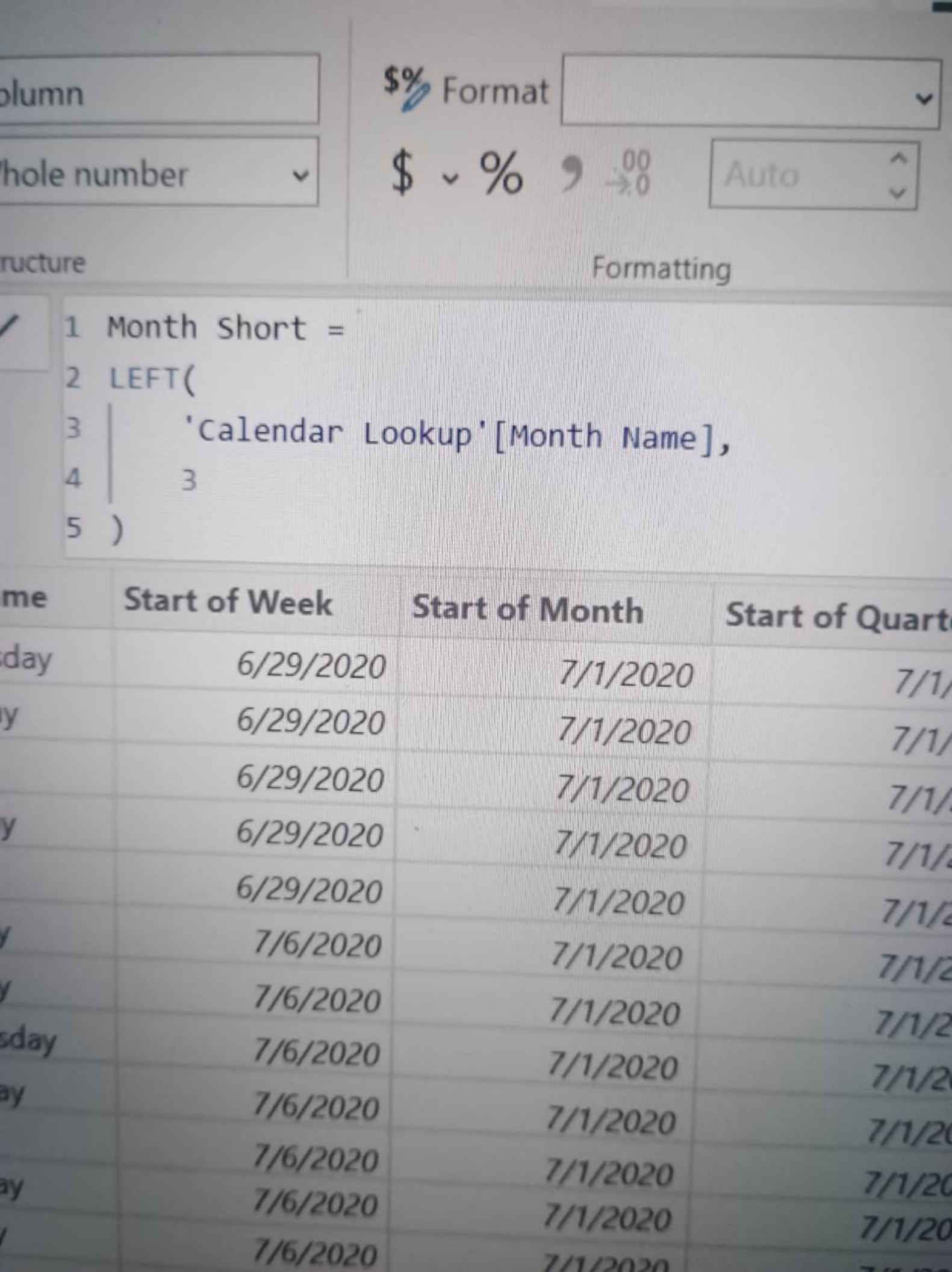
Properties

Sort

```
1 Customer Full Name (CC) =
2 'Customer Lookup'[Prefix] & " " & 'Customer Lookup'[FirstName] & " " & 'Customer Lookup'[LastName]
```

er	Full Name	Domain Name	Is Parent ?	Customer Priority	Income Level	Education Cate
	Mr. Shannon Carlson	Adventure Works	No	Standard	Average	Undergrad
	Mr. Jessie Liu	Adventure Works	No	Standard	Average	Undergrad
	Mr. Ruben Kapoor	Adventure Works	No	Standard	Average	Undergrad
	Mr. Ruben Muñoz	Adventure Works	No	Standard	Average	Undergrad
	Mr. Joe Rana	Adventure Works	No	Standard	Average	Undergrad
	Mr. Jarrod Suri	Adventure Works	No	Standard	Average	Undergrad
	Mr. Dustin Goldstein	Adventure Works	No	Standard	Average	Undergrad
	Mr. Clayton Jai	Adventure Works	No	Standard	Average	Undergrad
	Mr. Irving Schmidt	Adventure Works	No	Standard	Average	Undergrad
	Mr. Alan Huang	Adventure Works	No	Standard	Average	Undergrad
	Mr. Brendan Raji	Adventure Works	No	Standard	Average	Undergrad
	Mr. Gregory Becker	Adventure Works	No	Standard	Average	Undergrad
	Mr. Marco Vance	Adventure Works	No	Standard	Average	Undergrad
	Mr. Alejandro Hu	Adventure Works	No	Standard	Average	Undergrad
	Mr. Shane Fernandez	Adventure Works	No	Standard	Average	Undergrad
	Mr. Jay Raman	Adventure Works	No	Standard	Average	Undergrad


tegory
Customer Full Name (CC)
Mr. Shannon Carlson
Mr. Jessie Liu
Mr. Ruben Kapoor
Mr. Ruben Muñoz
Mr. Joe Rana
Mr. Jarrod Suri
Mr. Jarrod Suri
Mr. Clayton Jai
Mr. Irving Schmidt
Mr. Alan Huang
Mr. Brendan Raji
Mr. Gregory Becker
Mr. Marco Vance
Mr. Alejandro Hu
Mr. Shane Fernandez
Mr. Jay Raman
Mr. Damien Ye
Mr. Roy Martinez
Mr. Pedro Rana
Mr. Eugene Liang
Mr. Johnny Anand
Search
Measure Table
Calendar Lookup
Customer Lookup
AnnualIncome
BirthDate
Customer Full Name
Customer Priority
CustomerKey
Domain Name
Education Category
EducationLevel
EmailAddress
FirstName
Full Name
Gender
HomeOwner
Income Level
Is Parent ?


olumn
Format
hole number
$ % 9 .00 →.0
Auto
ructure
Formatting
1 Month Short =
2 LEFT(
3 'Calendar Lookup'[Month Name],
4 3
5)
me
Start of Week
Start of Month
Start of Quart
6/29/2020 7/1/2020
6/29/2020 7/1/2020
6/29/2020 7/1/2020
6/29/2020 7/1/2020
6/29/2020 7/1/2020
7/6/2020 7/1/2020
7/6/2020 7/1/2020
7/6/2020 7/1/2020
7/6/2020 7/1/2020
7/6/2020 7/1/2020
7/6/2020 7/1/2020
7/6/2020 7/1/2020

NEW MESSAGE

From: **Dianne A. Xu** *(Senior Analyst)*

Subject: **Couple random requests**

Good morning!

Hoping you can help with a couple quick updates to the model:

1) Ethan wants to make the month abbreviations ALL CAPS to make them more readable in our reports.

2) The product team asked us to break out the SKU category into its own field, which we can define as any characters before the first hyphen ("-") in the ProductSKU column.

Thanks, reach out with any questions!

Reply Forward

Key Objectives

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

2. 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

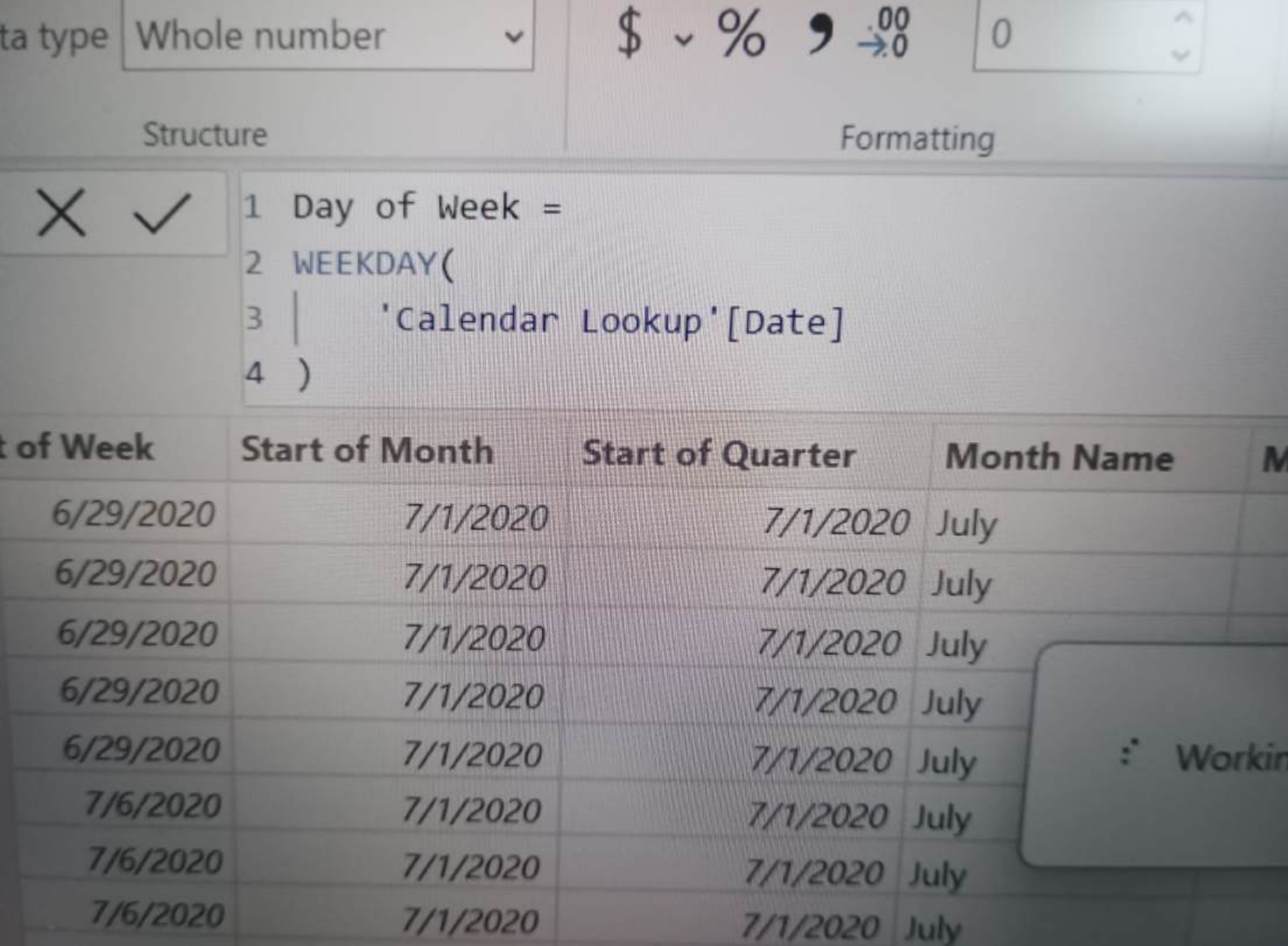
Solution Preview

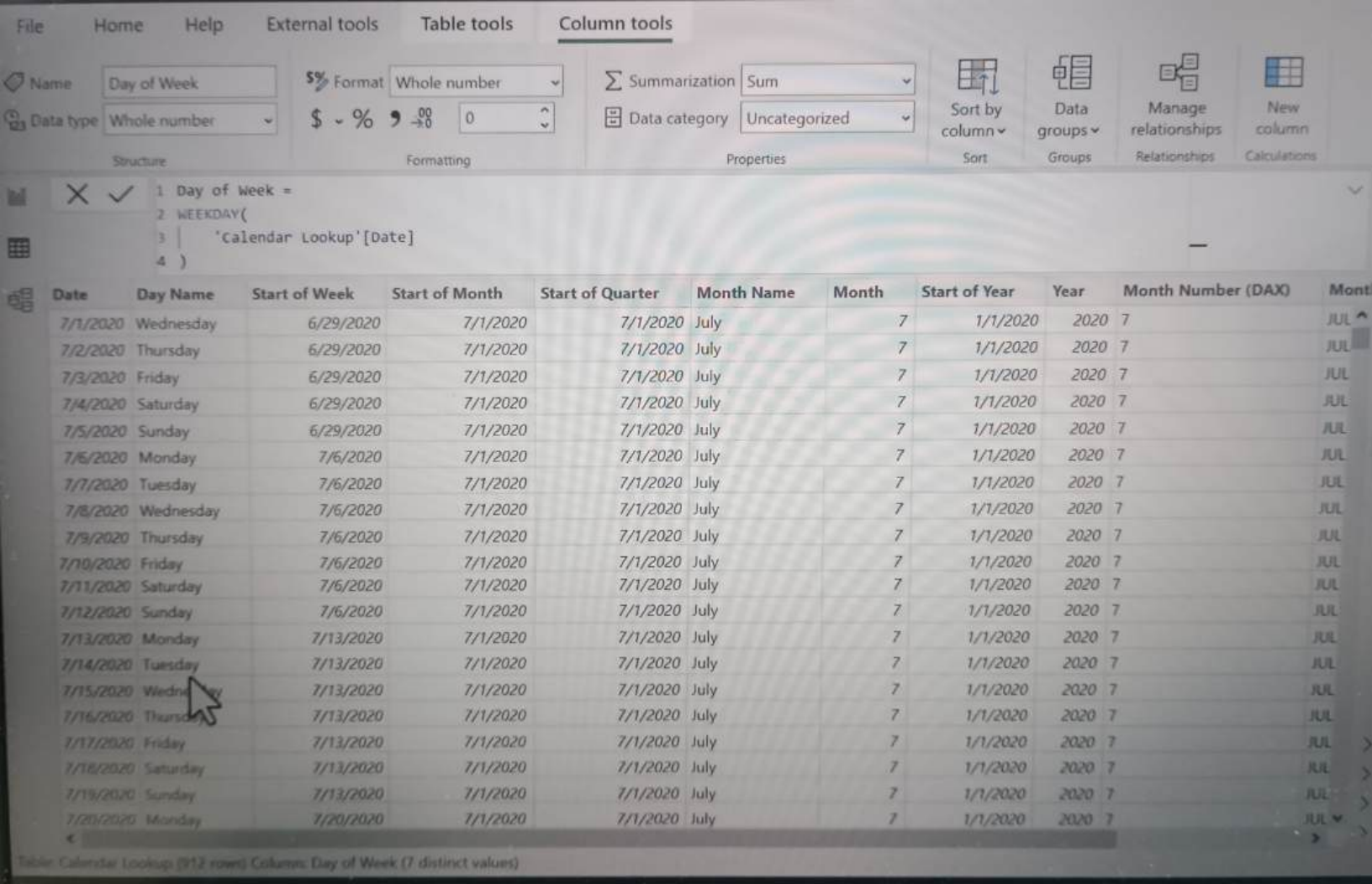
```
Month Short =
UPPER(
    LEFT(
        'Calendar Lookup'[Month Name],
        3
    )
)
```

```
SKU Category =
LEFT(
    'Product Lookup'[Product SKU],
    SEARCH(
        "-",
        'Product Lookup'[Product SKU]
    )
    -1
)
```

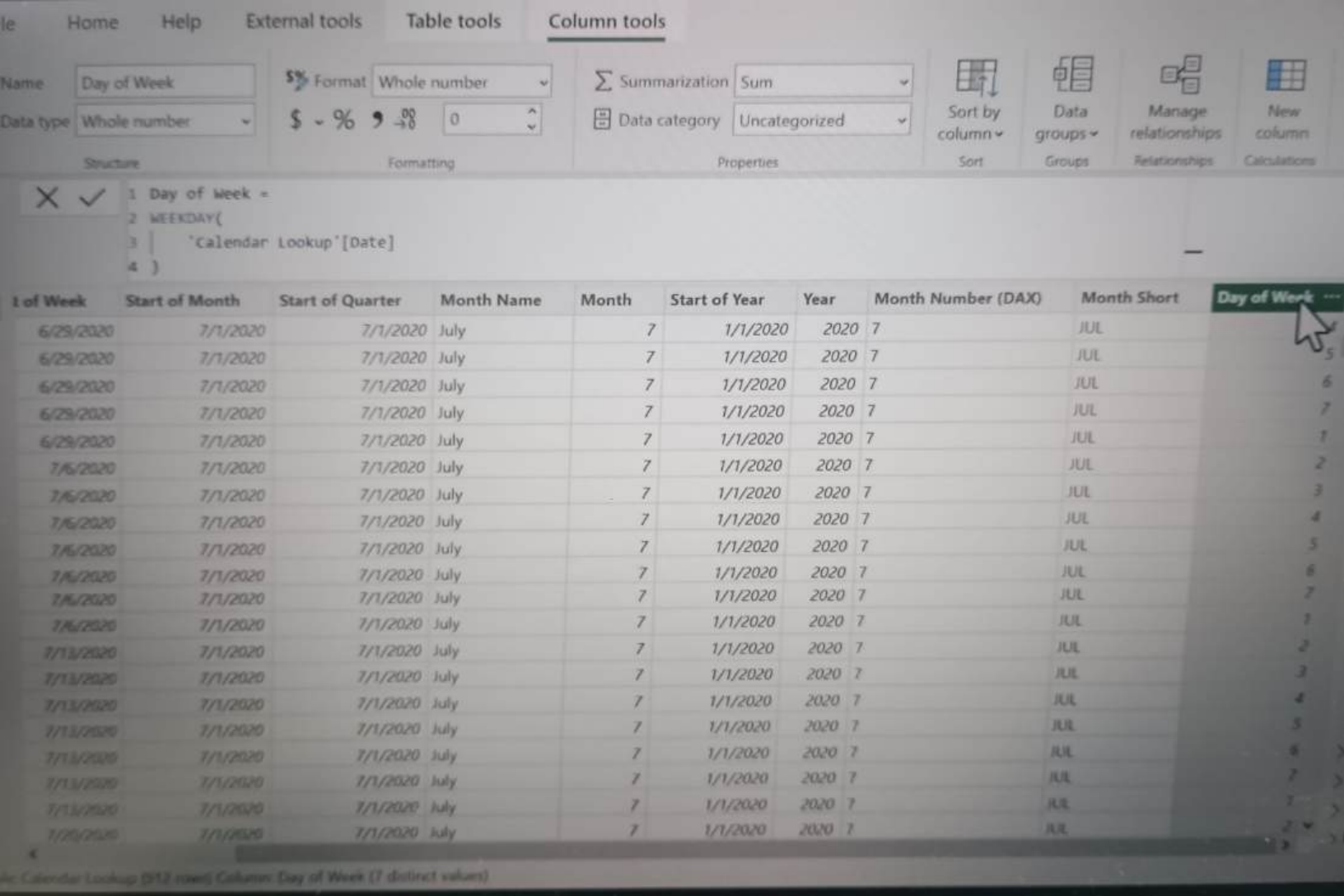
BASIC DATE & TIME FUNCTIONS

TODAY/NOW	Returns the current date or exact time	=**TODAY/NOW**()
DAY/MONTH/YEAR	Returns the day of the month (1-31), month of the year (1-12), or year of a given date	=**DAY/MONTH/YEAR**(Date)
HOUR/MINUTE/ SECOND	Returns the hour (0-23), minute (0-59), or second (0-59) of a given datetime value	=**HOUR/MINUTE/SECOND**(Datetime)
WEEKDAY/ WEEKNUM	Returns a weekday number from 1 (Sunday) to 7 (Saturday), or the week # of the year	=**WEEKDAY/WEEKNUM**(Date, *[ReturnType]*)
EOMONTH	Returns the date of the last day of the month, +/- a specified number of months	=**EOMONTH**(StartDate, Months)
DATEDIFF	Returns the difference between two dates, based on a given interval (day, hour, year, etc.)	=**DATEDIFF**(Date1, Date2, Interval)


ta type
Whole number
$ % , .00 .0
0
Structure
Formatting
1 Day of Week =
2 WEEKDAY(
3 'Calendar Lookup'[Date]
4)
t of Week
Start of Month
Start of Quarter
Month Name
6/29/2020 7/1/2020 7/1/2020 July
6/29/2020 7/1/2020 7/1/2020 July
6/29/2020 7/1/2020 7/1/2020 July
6/29/2020 7/1/2020 7/1/2020 July
6/29/2020 7/1/2020 7/1/2020 July
7/6/2020 7/1/2020 7/1/2020 July
7/6/2020 7/1/2020 7/1/2020 July
7/6/2020 7/1/2020 7/1/2020 July
Workin



Date	Day Name	Start of Week	Start of Month	Start of Quarter	Month Name	Month	Start of Year	Year	Month Number (DAX)	Mont
7/1/2020	Wednesday	6/29/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL
7/2/2020	Thursday	6/29/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL
7/3/2020	Friday	6/29/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL
7/4/2020	Saturday	6/29/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL
7/5/2020	Sunday	6/29/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL
7/6/2020	Monday	7/6/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL
7/7/2020	Tuesday	7/6/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL
7/8/2020	Wednesday	7/6/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL
7/9/2020	Thursday	7/6/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL
7/10/2020	Friday	7/6/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL
7/11/2020	Saturday	7/6/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL
7/12/2020	Sunday	7/6/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL
7/13/2020	Monday	7/13/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL
7/14/2020	Tuesday	7/13/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL
7/15/2020	Wednesday	7/13/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL
7/16/2020	Thursday	7/13/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL
7/17/2020	Friday	7/13/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL
7/18/2020	Saturday	7/13/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL
7/19/2020	Sunday	7/13/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL
7/20/2020	Monday	7/20/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL



le Home Help External tools Table tools Column tools

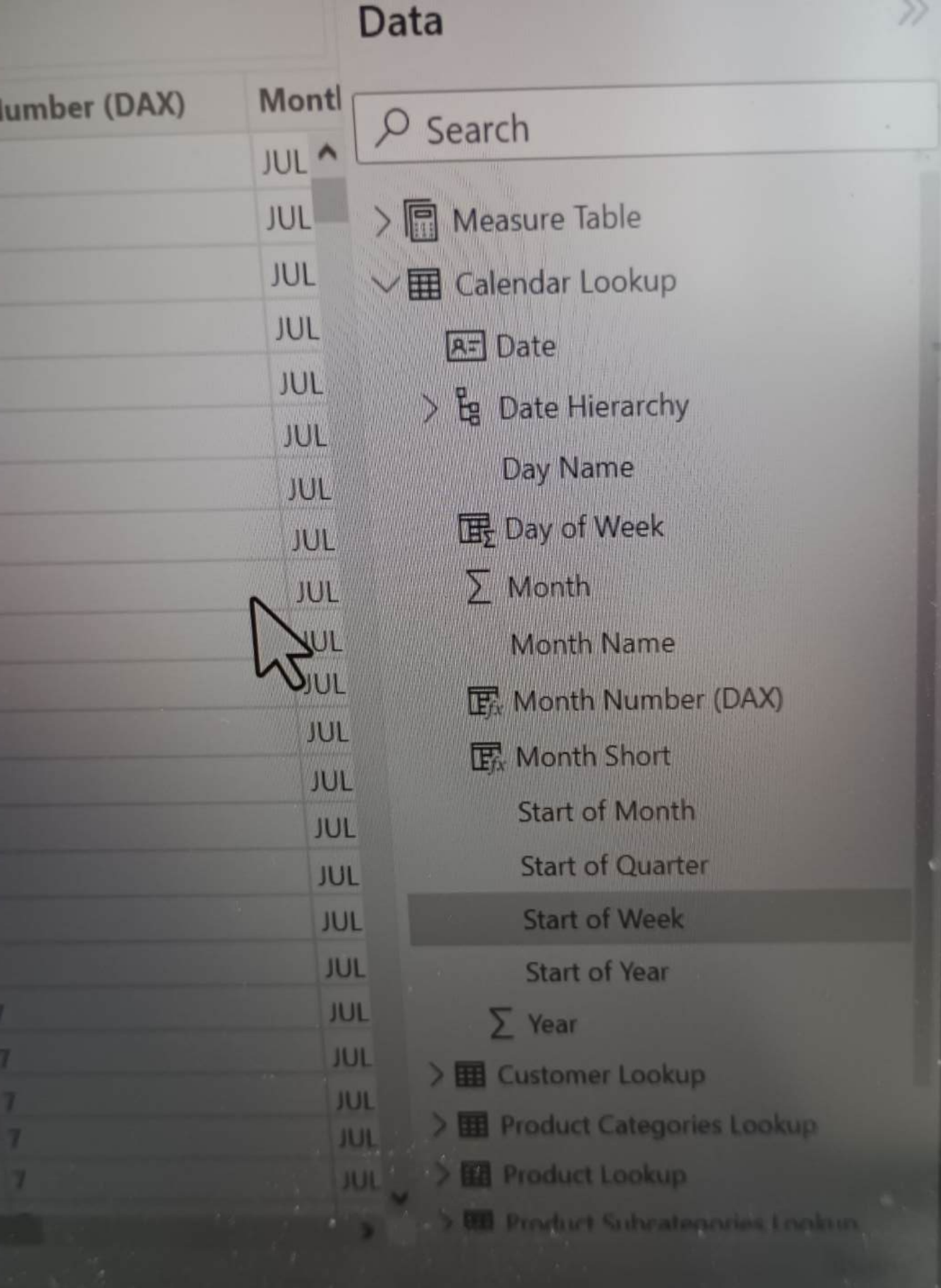
Name: Day of Week
Data type: Whole number
Format: Whole number
Summarization: Sum
Data category: Uncategorized
Sort by column
Data groups
Manage relationships
New column

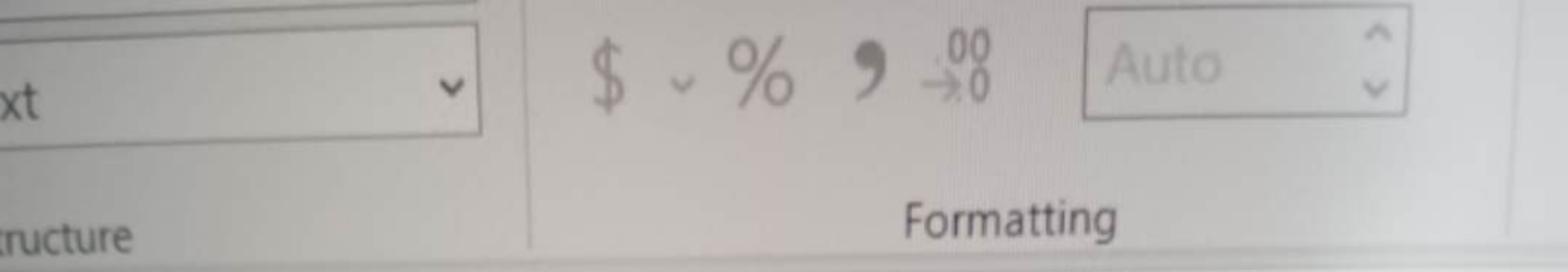
Structure | Formatting | Properties | Sort | Groups | Relationships | Calculations

```
1 Day of Week =
2 WEEKDAY(
3     'Calendar Lookup'[Date]
4 )
```

t of Week	Start of Month	Start of Quarter	Month Name	Month	Start of Year	Year	Month Number (DAX)	Month Short	Day of Week
6/29/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL	
6/29/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL	5
6/29/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL	6
6/29/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL	7
6/29/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL	1
7/6/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL	2
7/6/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL	3
7/6/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL	4
7/6/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL	5
7/6/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL	6
7/6/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL	7
7/6/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL	1
7/13/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL	2
7/13/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL	3
7/13/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL	4
7/13/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL	5
7/13/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL	6
7/13/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL	7
7/13/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL	1
7/20/2020	7/1/2020	7/1/2020	July	7	1/1/2020	2020	7	JUL	2

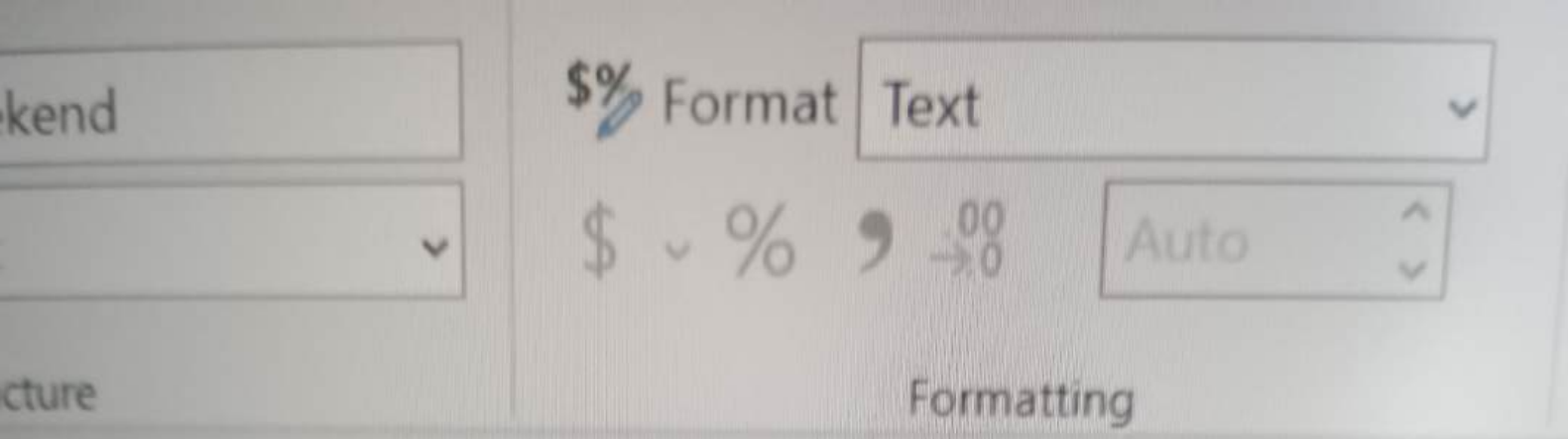
Calendar Lookup (912 rows) Column: Day of Week (7 distinct values)


Data
lumber (DAX)
Montl
JUL
Search
Measure Table
Calendar Lookup
Date
Date Hierarchy
Day Name
Day of Week
Month
Month Name
Month Number (DAX)
Month Short
Start of Month
Start of Quarter
Start of Week
Start of Year
Year
Customer Lookup
Product Categories Lookup
Product Lookup




```
1 Weekend =
2 IF(
3     'Calendar Lookup'[Day of Week] = 6 ||
4     'Calendar Lookup'[Day of Week] = 7,
5     "Weekend",
6     "Weekday"
7 )
```

of Month	Start of Quarter	Month Name	Month	Star
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	




```
1 Weekend =
2 IF(
3     'Calendar Lookup'[Day of Week] IN {6,7},
4     "Weekend",
5     "Weekday"
6 )
```

onth	Start of Quarter	Month Name	Month	Star
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	
7/1/2020	7/1/2020	July	7	

NEW MESSAGE

From: **Dianne A. Xu** *(Senior Analyst)*

Subject: **Customer birth years**

Hey there, super easy one for you.

The customer segmentation project got me wondering if there are any interesting patterns or insights based on customer age.

Could you please add a field in our customer table to extract only the year from the birthdate field?

Thanks!
-Dianne

Reply Forward

Notes Announcements Reviews Learning tools

Solution Preview

```
1 Birth Year =
2 YEAR(
3     'Customer Lookup'[BirthDate]
4 )
```