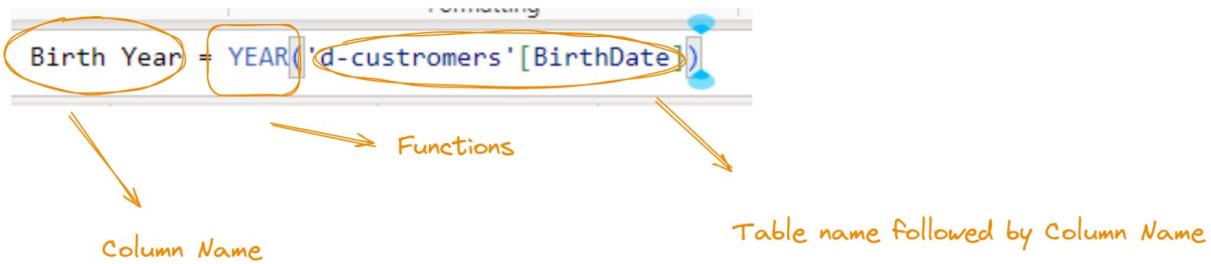


DAX - P2**Q1 -**

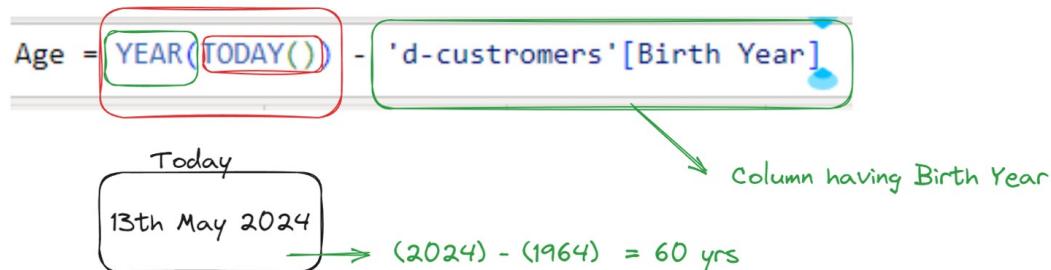
- Create a Calculated Column to extract the year part from the birth date column.

**BASIC DATE & TIME FUNCTIONS**

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

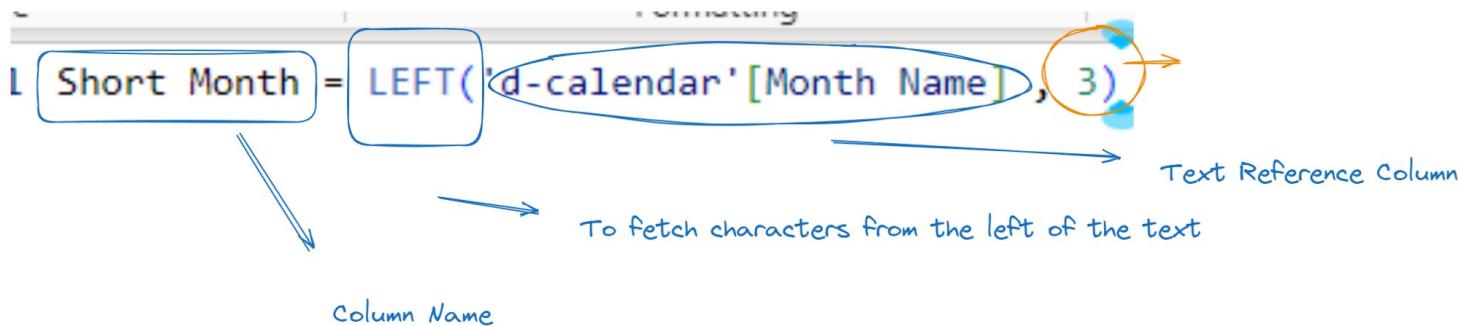
Q2 -

- Create a calculated column to extract the year part from the birth date column and another column extracting the current year to calculate the age of each customer.

**Q3 -**

- Create a calculated column to extract the first three letters from the month name column from the calendar table

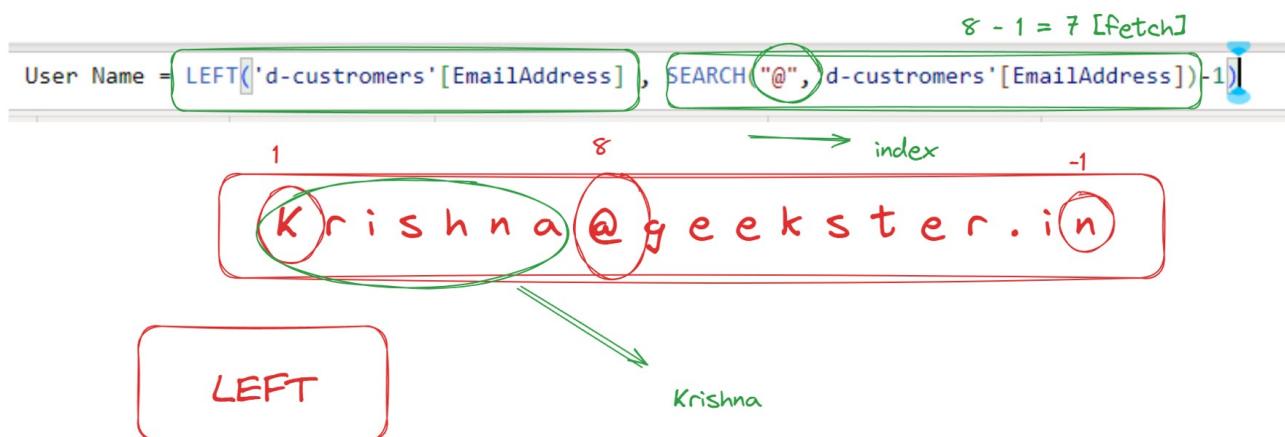
Create a calculated column to extract the first three letters from the month name column from the calendar table.



Q4 -

Create a calculated column to extract username from email id.

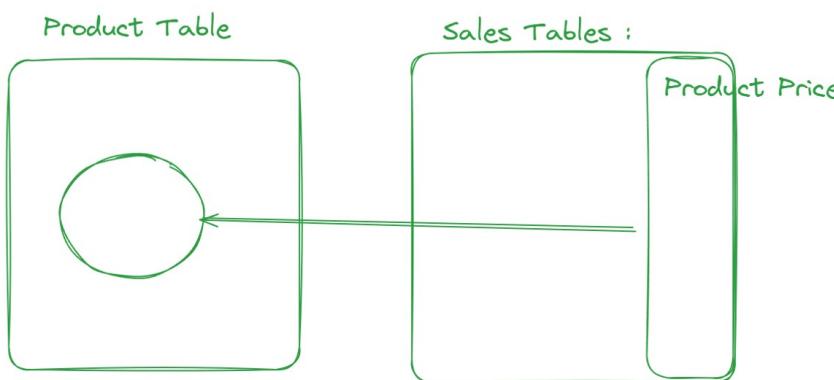
Instead of Find , We have Search()



Q5 -

Create a calculated column to pull the "product price" column from the "products" table to the "sales" table

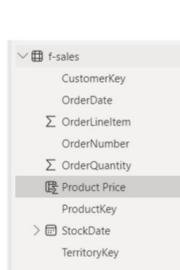
Model View - Relationship



RELATED → [Relationship]

StockDate	OrderNumber	ProductKey	CustomerKey	TerritoryKey	OrderLineItem	OrderQuantity	Product Price
15	03 June 2002	SO46718	360	12570	9	1	\$2,050
15	22 April 2002	SO46736	360	12341	9	1	\$2,050
15	05 May 2002	SO46776	360	12356	9	1	\$2,050
15	22 June 2002	SO46808	360	12347	9	1	\$2,050
15	11 May 2002	SO46826	360	12575	9	1	\$2,050
15	21 April 2002	SO47075	360	12685	9	1	\$2,050
15	01 May 2002	SO47098	360	12667	9	1	\$2,050
15	21 April 2002	SO47149	360	12669	9	1	\$2,050
15	04 June 2002	SO47212	360	12580	9	1	\$2,050
15	29 June 2002	SO47302	360	12670	9	1	\$2,050

→ Product Price = New Column Name
 Related Keyword - is used to apply or to pull columns from Related Table.
 d-products - is the Table Name
 ProductPrice is the Column that we are fetching.



Pro Tip :

You already working in the Sales Table , And then apply Related to fetch any column from Different Table . If they have Relationship in the model View.

Q6 -

Create a calculated column "Revenue" from "f-sales" using "order quantity" and "product price".

Revenue = 'f-sales'[OrderQuantity] * 'f-sales'[Product Price]

The same solution can be done with the help of using Product Price from Product Table instead of Sales Table

A screenshot of a Power BI report view. At the top, a DAX formula is shown: `Product Revenue = 'f-sales'[OrderQuantity] * RELATED('d-products'[ProductPrice])`. Below the formula is a table with columns: StockDate, OrderNumber, ProductKey, CustomerKey, TerritoryKey, OrderLineItem, OrderQuantity, Product Price, Revenue, and Product Revenue. The table contains five rows of data. Orange arrows point from the formula to the 'OrderQuantity' column in the table, and another arrow points from the formula to the 'Product Price' column. Labels 'New Column Name' and 'Order Quantity from Sales Table * Product Price from Product Table' are placed near the arrows.

	StockDate	OrderNumber	ProductKey	CustomerKey	TerritoryKey	OrderLineItem	OrderQuantity	Product Price	Revenue	Product Revenue
2015	03 June 2002	SO46718	360	12570	9	1	1	\$2,050	\$2,050	₹ 2,050
2015	22 April 2002	SO46736	360	12341	9	1	1	\$2,050	\$2,050	₹ 2,050
2015	05 May 2002	SO46776	360	12356	9	1	1	\$2,050	\$2,050	₹ 2,050
2015	22 June 2002	SO46808	360	12347	9	1	1	\$2,050	\$2,050	₹ 2,050
2015	11 May 2002	SO46826	360	12575	9	1	1	\$2,050	\$2,050	₹ 2,050

Measure - Report View

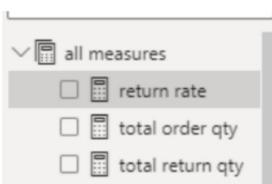
Q7 -

Create a measure to calculate the return rate of each subcategory.

1 . Return Rate - To calculate Return Rate ,

$$\frac{\text{Total Return Return}}{\text{Total Order Quantity}} = \frac{10}{1000} = 0.01 \% \text{ Return Rate}$$

Div / 0 = Error [0]

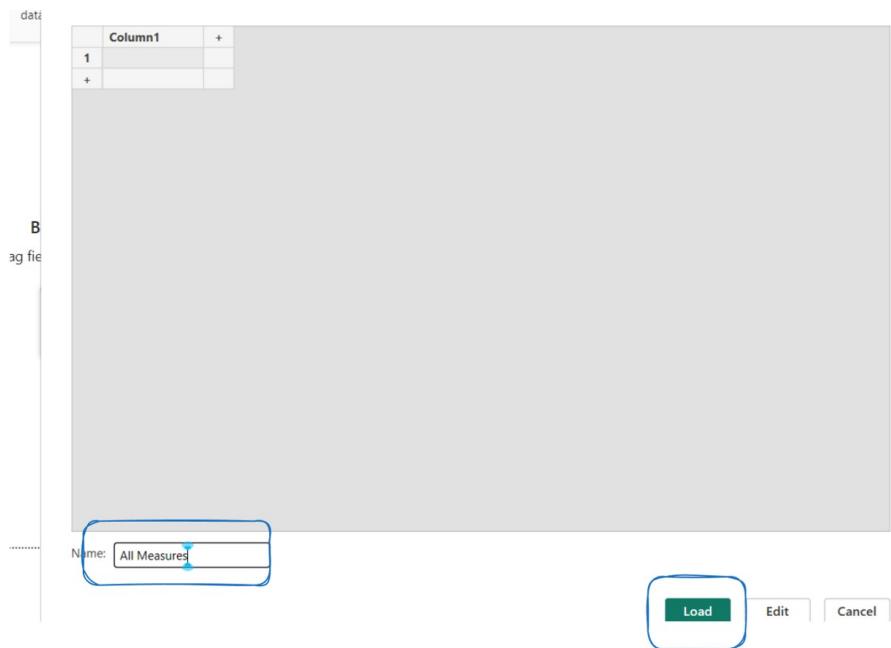


We are not working on a specific Table to get the result.
→ New Measure Table to store all the measure Value.

→ Steps to Create a all measure Table :

Step 1 : To create a new Empty Table and Rename it to "All Measures". & Load the data.



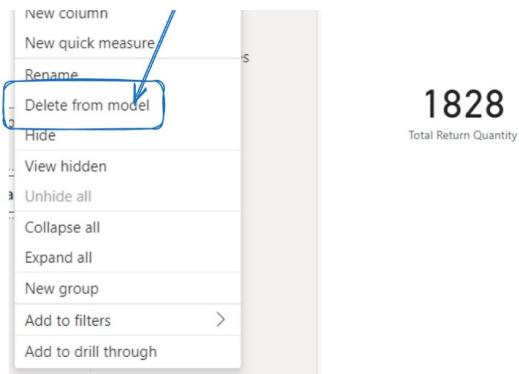


Step 2 - Try to Create a New Measure : And Delete the Empty Column to convert that particular Table as a specific Table for Calculating Measure.

New Measure :

Measure | Formatting |
1 Total Return Quantity = SUM('f>Returns'[ReturnQuantity])

Step 3 : Try to delete empty column in order to convert this table into Measure Format.



1828

Total Return Quantity

Q7-

Total Return Quantity , Total Order Quantity , Return Rate.

1 Total Return Quantity = `SUM('f>Returns'[ReturnQuantity])`

1 Total Order Quantity = `SUM('f-Sales'[OrderQuantity])`

1 Return Rate = `DIVIDE([Total Return Quantity] , [Total Order Quantity] , 0)`

Category Name	Total Order Quantity	Total Return Quantity	Return Rate
Accessories	57809	1130	0.02
Bikes	13929	429	0.03
Clothings	12436	269	0.02
Total	84174	1828	0.02



We can use measure to showcase as an individual score card.

SubcategoryName	Total Order Quantity	Total Return Quantity	Return Rate
Gloves	2644	49	0.02
Helmets	6034	188	0.03
Hydration Packs	695	25	0.04
Jerseys	3113	93	0.03
Mountain Bikes	4706	136	0.03
Road Bikes	7099	223	0.03
Shorts	944	40	0.04
Socks	1063	22	0.02
Tires and Tubes	29772	534	0.02
Total	84174	1828	0.02

Country	Total Order Quantity	Total Return Quantity	Return Rate
Australia	17951	404	0.02
Canada	10894	238	0.02
France	7862	186	0.02
Germany	7950	163	0.02
United Kingdom	9694	204	0.02
United States	29823	633	0.02
Total	84174	1828	0.02

Or use it as based on some Related Data