Creating and Using Different Types of Calculations: Date Calculations

When manipulating **Date** fields, you can use the various **Date** functions supported by Tableau. At this point in time (that is, in version 2020.1), these are as follows:

- **DATENAME**, **DATEPART**, **DATETRUNC**, **YEAR**, **QUARTER**, **MONTH**, **WEEK**, **DAY**, **ISOYEAR**, **ISOQUARTER**, **ISOWEEK**, and **ISOWEEKDAY**, which can be used to find the date part of the **Date** field.

- **DATEDIFF** and **DATEADD**, used to find the difference between two dates or to generate a new **Date** field based on an incremental interval.

- **TODAY** and **NOW**, which give the current date or date and time.

- **ISDATE**, used to find out whether a given field is a **Date** field.

You will now use a **Date** calculation to find out how many months it has been since your customers last made a purchase.

Exercise 03: Creating a Date Calculation

The objective of this exercise is to create a **Date** calculation to find the number of months since the last purchase for your customers. You will continue using your **Orders** data from **Sample-Superstore.xlsx** and use the **Customer Name** and the **Order Date** fields.

Perform the following steps:

1. Start by dragging **Customer Name** into the Rows shelf. Then, right-click drag and drop the **Order Date** field into the **Rows** shelf, which should create a **Menu**. Select **MDY(Order Date)**. Refer to the following screenshot:
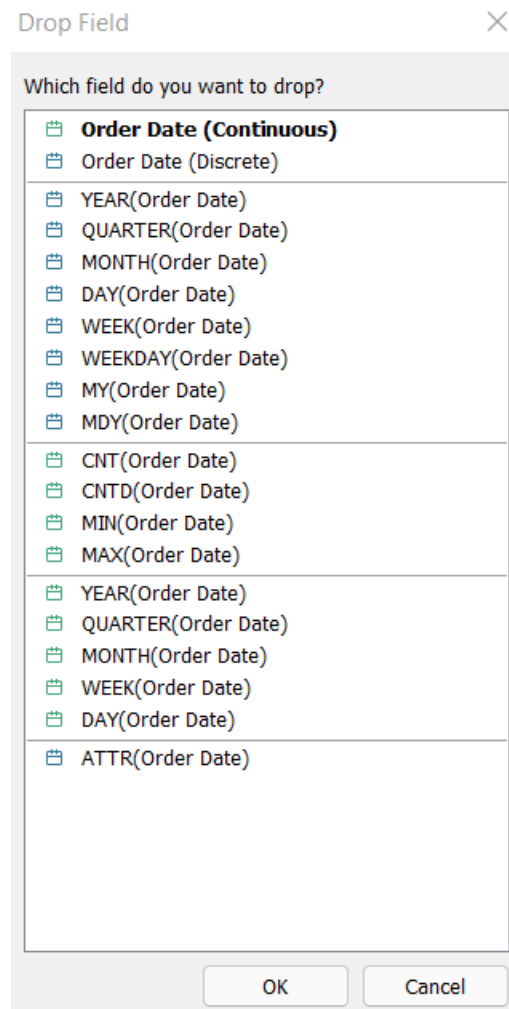

Figure 1: A screenshot showing the right-click drag-drop menu for Order Date

Now you can see all order dates at the customer level. There is no point looking at all transactional dates for every customer. You are only interested in the last purchase date, and how many months it has been since it occurred.

2. To achieve this, first create a calculation called **Last purchase date** with the following formula:

```
MAX([Order Date])
```

Refer to the following screenshot:

```
Last purchase date                                          ✕
────────────────────────────────────────────────────────────
MAX([Order Date])



                                                              ◀



The calculation is valid.                    [ Apply ]  [ OK ]
```

Figure 2: A screenshot showing the Last purchase date calculation

3. Since this calculation will be computed on the fly, the `Max` date is dependent on the dimensions in the view. If you drag and drop this new field into your `Rows` shelf, you should notice that the values are the same as for the `MDY(Order Date)`. This won't work for you; you want the Max date for each customer, and hence you must remove the `MDY(Order Date)` granularity. This will update your view, as shown in the following screenshot:

Figure 3: A screenshot showing the Last purchase date for each customer

Now you have your **Last purchase date** field, it is time to find out how many months it has been since the customers last made a purchase. This can be achieved by finding the difference between two dates, that is, **Last purchase date** and, ideally, **Today**. However, since your data is not daily-updating, you will consider the end date as December 31, 2019, which is the last date in the data.

4. Create a new calculated field called *Months since last purchase* and use the following formula:

```
DATEDIFF('month', [Last purchase date], #2019-12-31#)
```
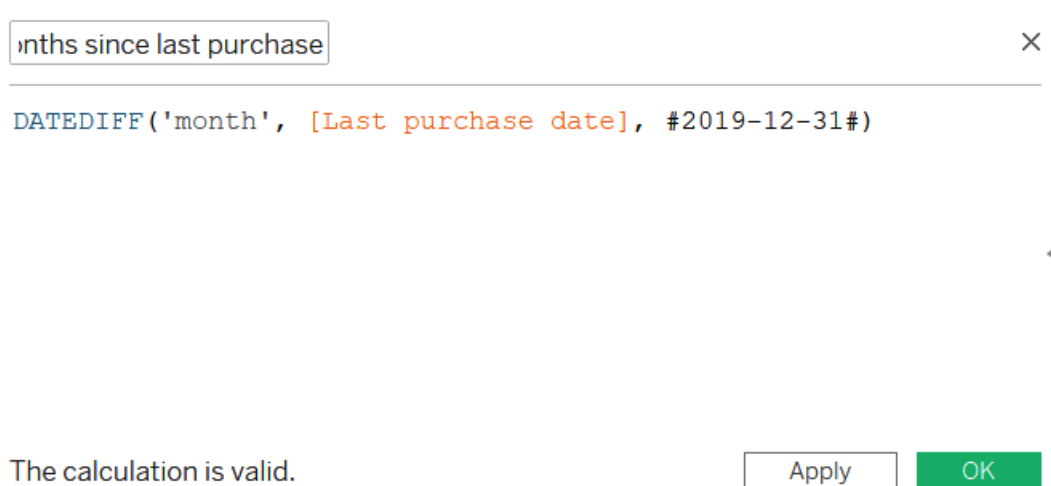
Refer to the following screenshot:

```
nths since last purchase                                          ×
────────────────────────────────────────────────────────────────
DATEDIFF('month', [Last purchase date], #2019-12-31#)




                                                              ◄

The calculation is valid.                      [ Apply ]    [ OK ]
```

Figure 4: A screenshot showing the Months since last purchase calculation

5. After you save this calculation, you can drag it into the **Text** shelf, and should get
    the desired output. This calculation finds the difference in months between **Last**
    purchase date and December 31, 2019. A point to remember is that when you
    need to enter a hardcoded date, it will start and end with a *hash* (**#**), as shown
    above. Further, if this data was daily-updating and you wanted to find the
    difference with respect to **Today**, that is, the current date, then you could use
    the **Today()** function, and the calculation would update as shown here:

```
DATEDIFF('month', [Last purchase date], Today())
```

Customer Name	Last purchase date	
Aaron Bergman	11/10/2018	13
Aaron Hawkins	12/18/2019	0
Aaron Smayling	10/3/2019	2
Adam Bellavance	11/6/2019	1
Adam Hart	11/26/2019	1
Adam Shillingsburg	12/2/2019	0
Adrian Barton	11/19/2019	1
Adrian Hane	10/31/2019	2
Adrian Shami	11/19/2019	1
Aimee Bixby	11/19/2019	1
Alan Barnes	12/5/2019	0
Alan Dominguez	12/1/2019	0
Alan Haines	11/5/2019	1
Alan Hwang	12/24/2019	0
Alan Schoenberger	12/8/2019	0
Alan Shonely	6/21/2019	6
Alejandro Ballentine	7/17/2019	5
Alejandro Grove	3/27/2019	9
Alejandro Savely	12/11/2019	0
Aleksandra Gannaway	9/3/2019	3
Alex Avila	6/29/2019	6
Alex Grayson	12/22/2019	0
Alex Russell	9/10/2019	3
Alice McCarthy	12/2/2019	0
Allen Armold	12/11/2019	0
Allen Goldenen	9/7/2019	3
Allen Rosenblatt	8/27/2019	4
Alyssa Crouse	5/12/2019	7
Alyssa Tate	12/9/2019	0
Amy Cox	12/19/2019	0
Amy Hunt	5/14/2019	7
Andrew Allen	4/15/2019	8

Figure 5: A screenshot showing the final output of the Date calculation