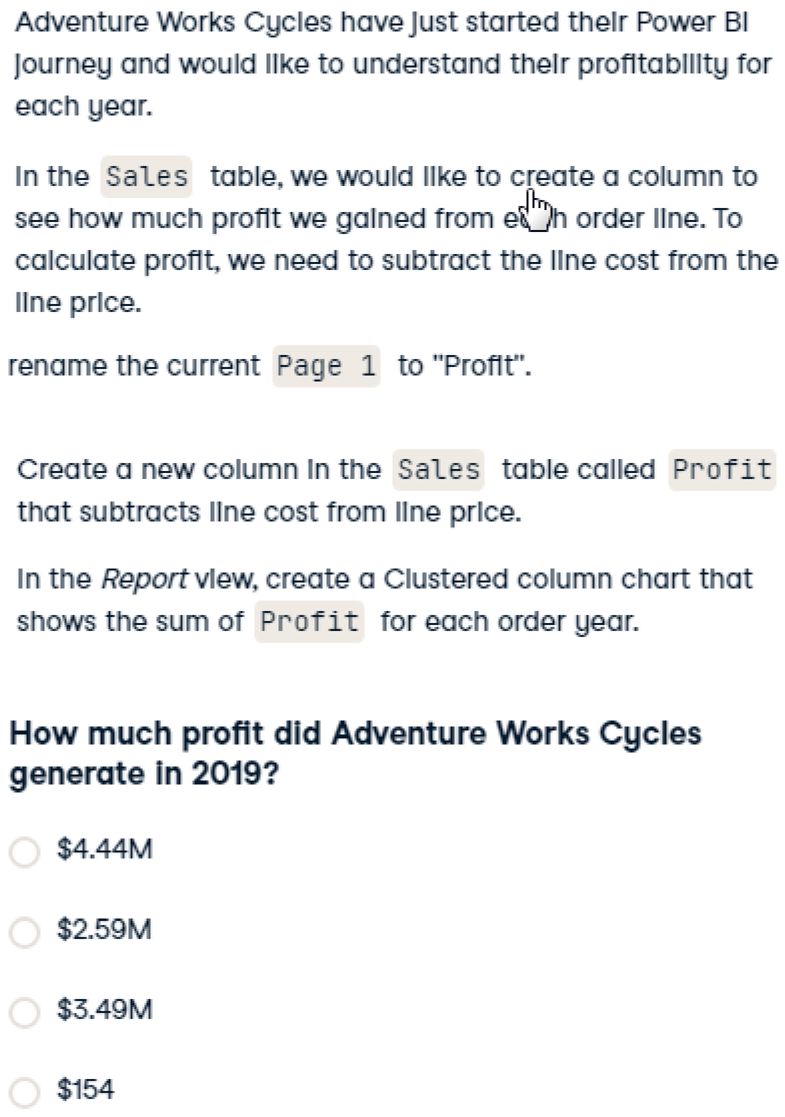
Labs:

DAX



Adventure Works Cycles have just started their Power BI journey and would like to understand their profitability for each year.

In the `Sales` table, we would like to create a column to see how much profit we gained from each order line. To calculate profit, we need to subtract the line cost from the line price.

rename the current `Page 1` to "Profit".

Create a new column in the `Sales` table called `Profit` that subtracts line cost from line price.

In the *Report* view, create a Clustered column chart that shows the sum of `Profit` for each order year.

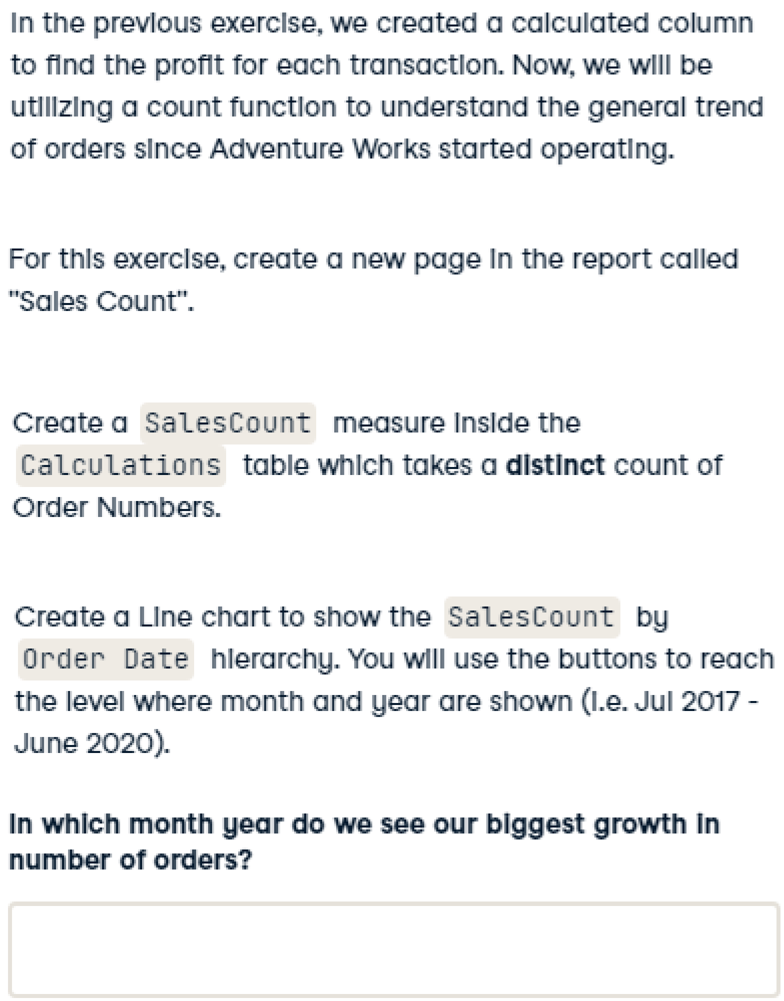
How much profit did Adventure Works Cycles generate in 2019?

○ $4.44M

○ $2.59M

○ $3.49M

○ $154



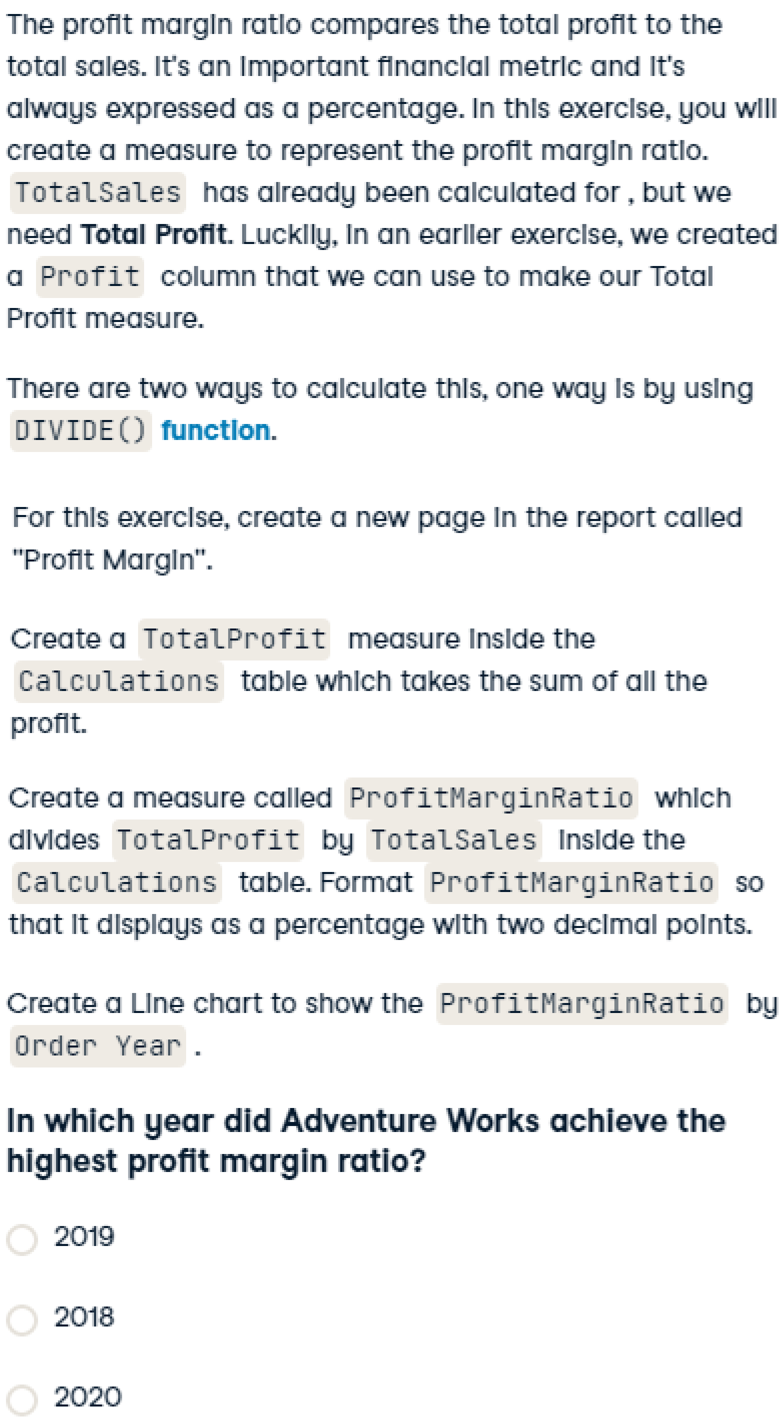
In the previous exercise, we created a calculated column to find the profit for each transaction. Now, we will be utilizing a count function to understand the general trend of orders since Adventure Works started operating.

For this exercise, create a new page in the report called "Sales Count".

Create a `SalesCount` measure inside the `Calculations` table which takes a **distinct** count of Order Numbers.

Create a Line chart to show the `SalesCount` by `Order Date` hierarchy. You will use the buttons to reach the level where month and year are shown (i.e. Jul 2017 - June 2020).

In which month year do we see our biggest growth in number of orders?



The profit margin ratio compares the total profit to the total sales. It's an important financial metric and it's always expressed as a percentage. In this exercise, you will create a measure to represent the profit margin ratio. `TotalSales` has already been calculated for , but we need **Total Profit**. Luckily, in an earlier exercise, we created a `Profit` column that we can use to make our Total Profit measure.

There are two ways to calculate this, one way is by using `DIVIDE()` **function**.

For this exercise, create a new page in the report called "Profit Margin".

Create a `TotalProfit` measure inside the `Calculations` table which takes the sum of all the profit.

Create a measure called `ProfitMarginRatio` which divides `TotalProfit` by `TotalSales` inside the `Calculations` table. Format `ProfitMarginRatio` so that it displays as a percentage with two decimal points.

Create a Line chart to show the `ProfitMarginRatio` by `Order Year` .

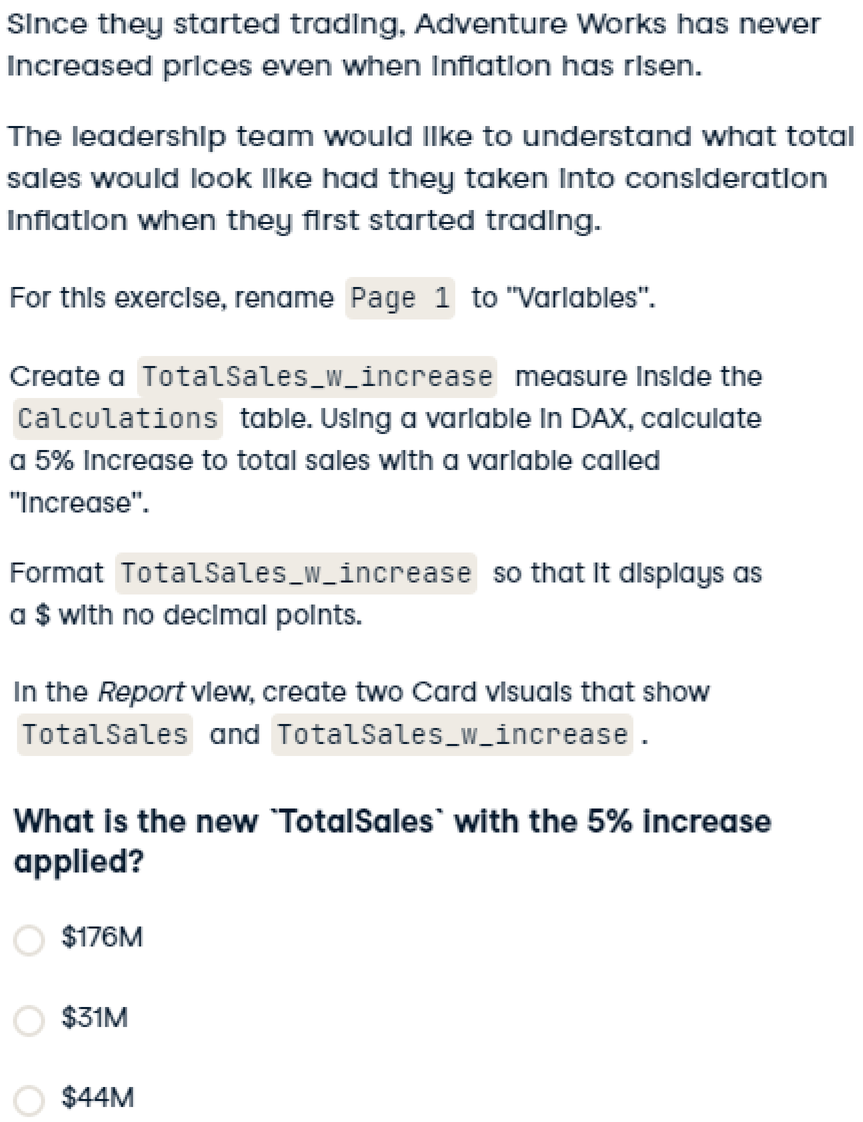
In which year did Adventure Works achieve the highest profit margin ratio?

○ 2019

○ 2018

○ 2020

Context in DAX Formulas



Since they started trading, Adventure Works has never increased prices even when inflation has risen.

The leadership team would like to understand what total sales would look like had they taken into consideration inflation when they first started trading.

For this exercise, rename `Page 1` to "Variables".

Create a `TotalSales_w_increase` measure inside the `Calculations` table. Using a variable in DAX, calculate a 5% increase to total sales with a variable called "Increase".

Format `TotalSales_w_increase` so that it displays as a $ with no decimal points.

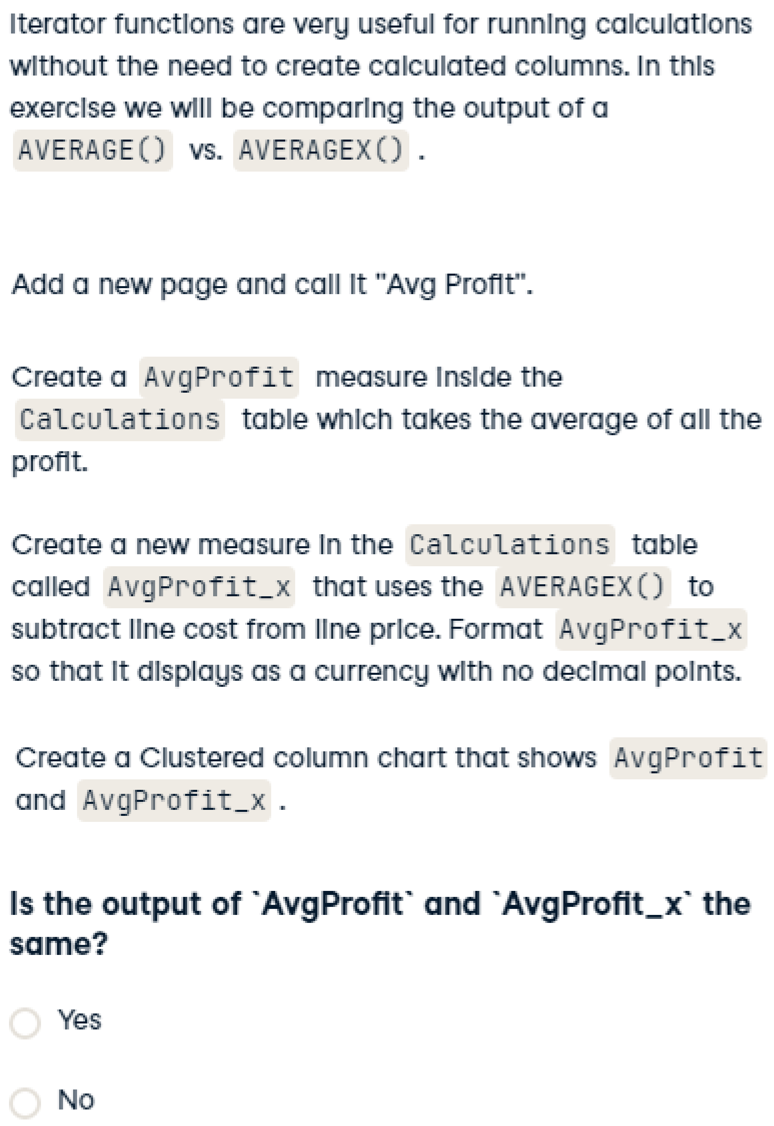
In the *Report* view, create two Card visuals that show `TotalSales` and `TotalSales_w_increase` .

What is the new `TotalSales` with the 5% increase applied?

○ $176M

○ $31M

○ $44M



Iterator functions are very useful for running calculations without the need to create calculated columns. In this exercise we will be comparing the output of a `AVERAGE()` vs. `AVERAGEX()` .

Add a new page and call it "Avg Profit".

Create a `AvgProfit` measure inside the `Calculations` table which takes the average of all the profit.

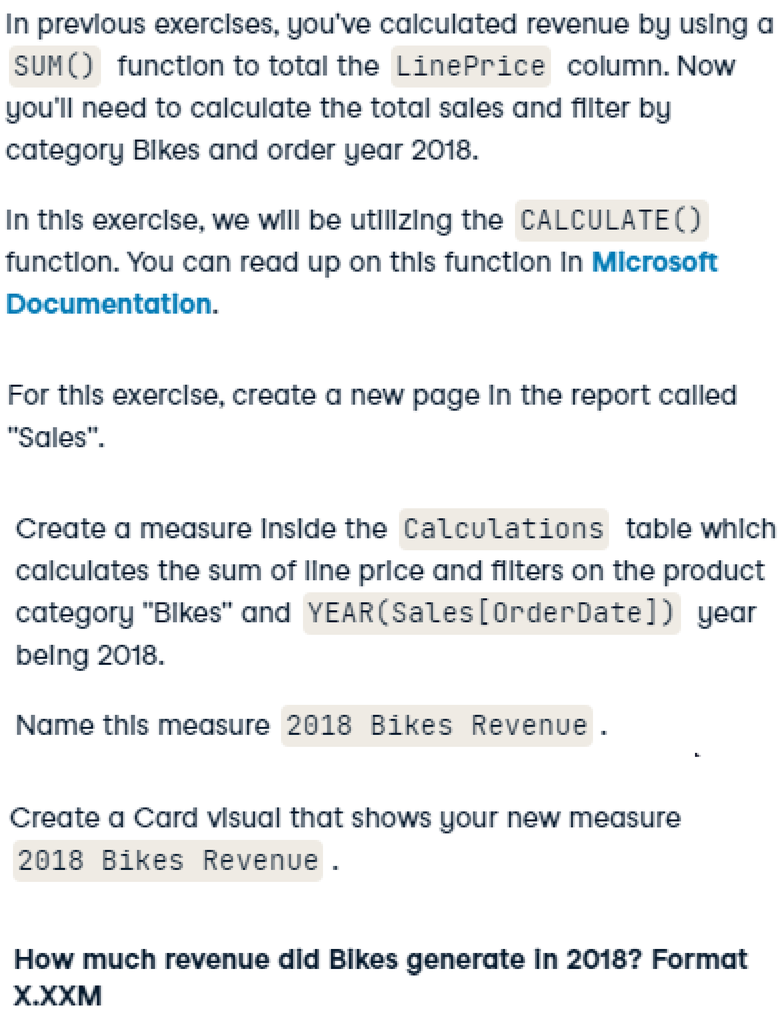
Create a new measure in the `Calculations` table called `AvgProfit_x` that uses the `AVERAGEX()` to subtract line cost from line price. Format `AvgProfit_x` so that it displays as a currency with no decimal points.

Create a Clustered column chart that shows `AvgProfit` and `AvgProfit_x` .

Is the output of `AvgProfit` and `AvgProfit_x` the same?

○ Yes

○ No



In previous exercises, you've calculated revenue by using a `SUM()` function to total the `LinePrice` column. Now you'll need to calculate the total sales and filter by category Bikes and order year 2018.

In this exercise, we will be utilizing the `CALCULATE()` function. You can read up on this function in **Microsoft Documentation**.

For this exercise, create a new page in the report called "Sales".

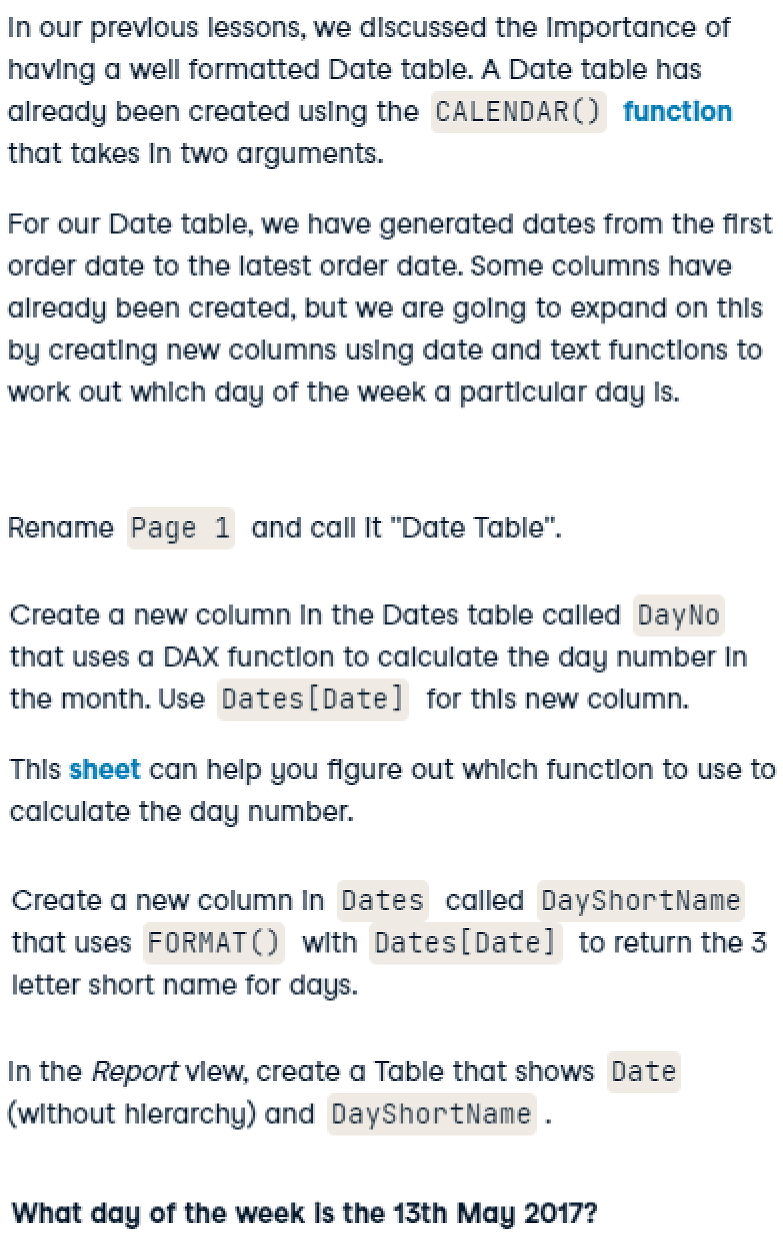
Create a measure inside the `Calculations` table which calculates the sum of line price and filters on the product category "Bikes" and `YEAR(Sales[OrderDate])` year being 2018.

Name this measure `2018 Bikes Revenue`.

Create a Card visual that shows your new measure `2018 Bikes Revenue`.

How much revenue did Bikes generate in 2018? Format X.XXM

Working with Dates



In our previous lessons, we discussed the importance of having a well formatted Date table. A Date table has already been created using the `CALENDAR()` **function** that takes in two arguments.

For our Date table, we have generated dates from the first order date to the latest order date. Some columns have already been created, but we are going to expand on this by creating new columns using date and text functions to work out which day of the week a particular day is.

Rename `Page 1` and call it "Date Table".

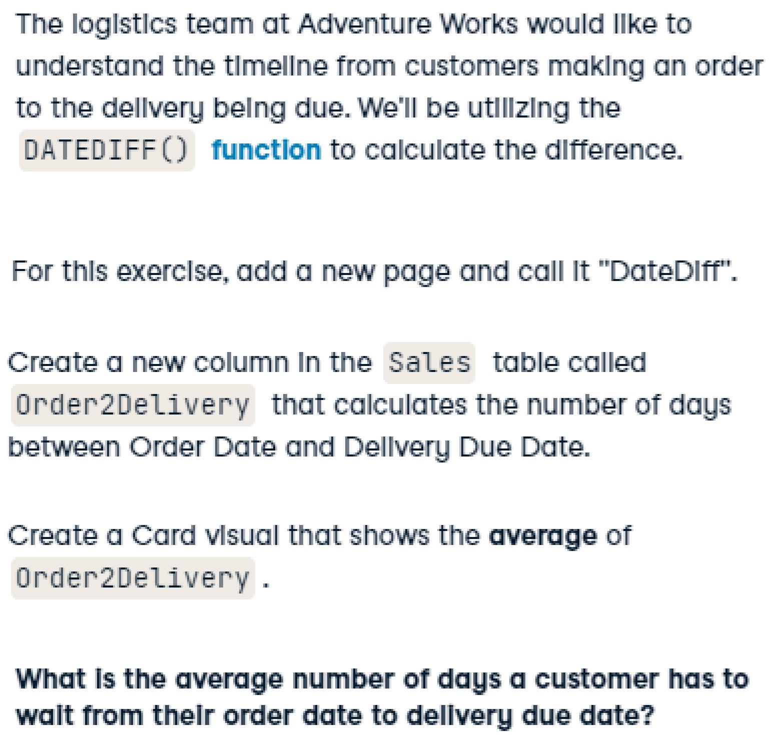
Create a new column in the Dates table called `DayNo` that uses a DAX function to calculate the day number in the month. Use `Dates[Date]` for this new column.

This **sheet** can help you figure out which function to use to calculate the day number.

Create a new column in `Dates` called `DayShortName` that uses `FORMAT()` with `Dates[Date]` to return the 3 letter short name for days.

In the *Report* view, create a Table that shows `Date` (without hierarchy) and `DayShortName` .

What day of the week is the 13th May 2017?



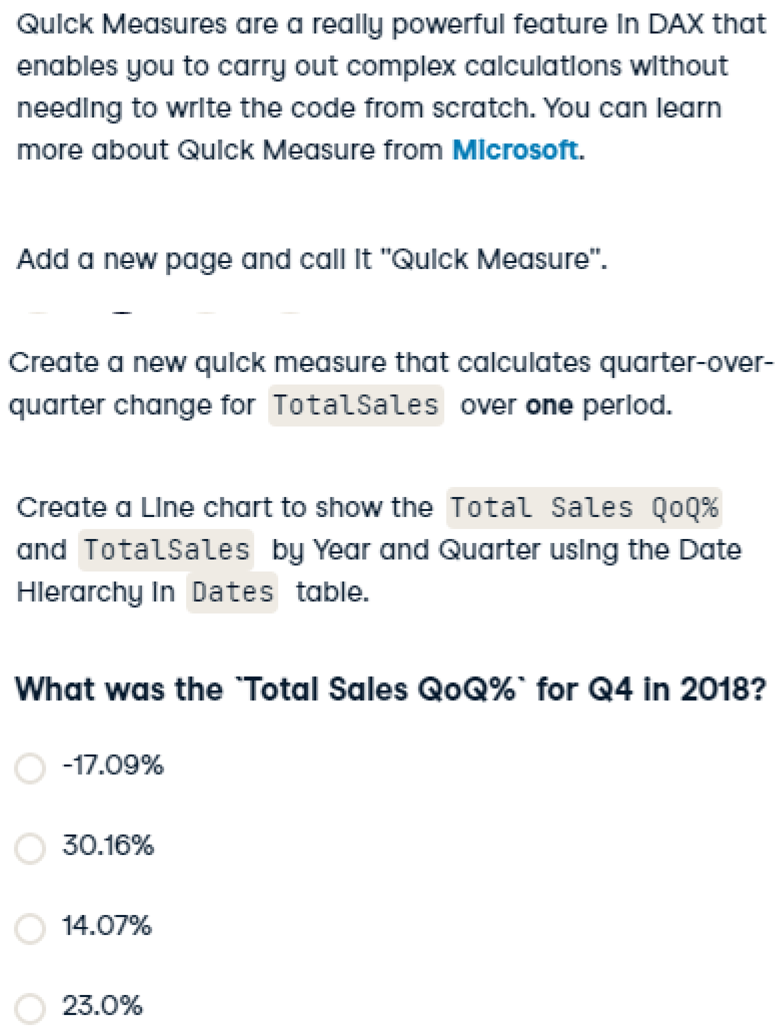
The logistics team at Adventure Works would like to understand the timeline from customers making an order to the delivery being due. We'll be utilizing the `DATEDIFF()` function to calculate the difference.

For this exercise, add a new page and call it "DateDiff".

Create a new column in the `Sales` table called `Order2Delivery` that calculates the number of days between Order Date and Delivery Due Date.

Create a Card visual that shows the **average** of `Order2Delivery`.

What is the average number of days a customer has to wait from their order date to delivery due date?



Quick Measures are a really powerful feature in DAX that enables you to carry out complex calculations without needing to write the code from scratch. You can learn more about Quick Measure from **Microsoft**.

Add a new page and call it "Quick Measure".

Create a new quick measure that calculates quarter-over-quarter change for `TotalSales` over **one** period.

Create a Line chart to show the `Total Sales QoQ%` and `TotalSales` by Year and Quarter using the Date Hierarchy in `Dates` table.

What was the `Total Sales QoQ%` for Q4 in 2018?

○ -17.09%

○ 30.16%

○ 14.07%

○ 23.0%

In the last exercise, you created a quick measure `Total Sales QoQ%` that calculated Quarter-over-quarter change in Total Sales.

The senior management team would like to understand year-over-year changes in number of sales, therefore, you need to create a copy of the `Total Sales QoQ%` measure and adjust it to calculate year-over-year changes.

Add a new page and call it "YoY"`.

In the `Calculations` table, review the DAX from the quick measure `Total Sales QoQ%` that you created in the last exercise.

Now duplicate this measure and rename it `Sales Count YoY%`

Adjust the query to calculate YoY growth for **SalesCount** and format as a percentage.

Create a Line chart that shows `Sales Count YoY%` by year using `Dates[Date]` hierarchy.

According to the graph you just created, what was the Sales Count YoY% for the 2019?