#### Generating passwords – As you already know, DAX have many functions including those that pertains to the manipulation of string, date, and numbers. For example, LEFT() returns the first specified number of characters of a text starting from the left. As you might have guessed, there is also RIGHT(), which returns the last specified number of characters of a text.

#### Close any open reports. Download Generating\_Passwords.pbix from Canvas as your seed project. Open it up and study the Model view. The ERD should look like the image below.

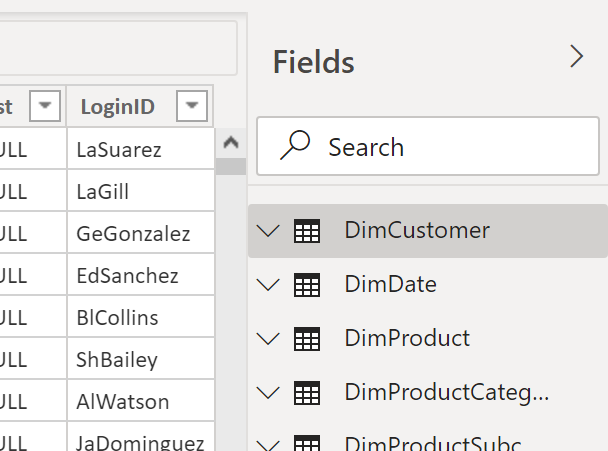
#### 

#### Go to Data view, and investigate DimCustomer table. Look at the last field (LoginID). Observe how the data on this column is a combination of the first two characters of the customer’s FirstName and concatenated by its LastName value. A DAX function called LEFT() which returns the first specified number of characters of a text was used and the & (ampersand) operator to connect it to the LastName column. The combination of the first two letters of its FirstName and the full LastName makes up a customer’s username.

#### Along with providing customers with a username, we want to provide a temporary password. We want something that is easy to remember but not too easy to guess for security reasons. In this exercise, you will generate a new calculated column called TempPassword by combining:

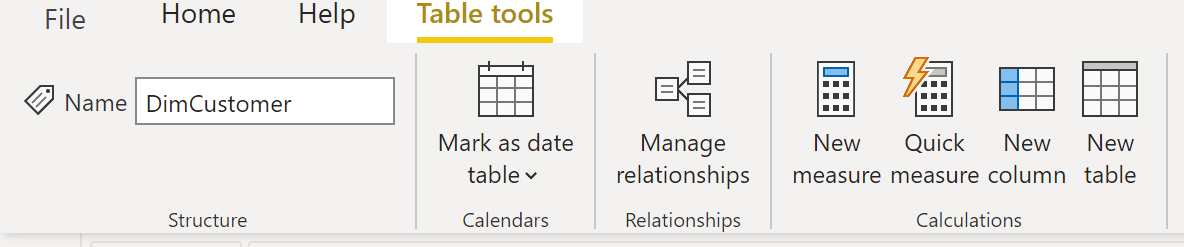
* The **last** four digits of a customer's phone number
* The customer's full year of birth (e.g, 1990)

While in Data view and DimCustomer is enabled:



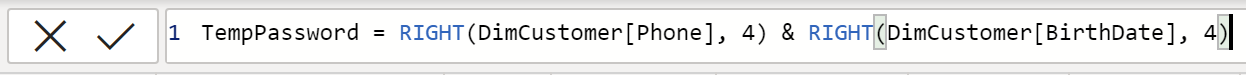


Click New Column



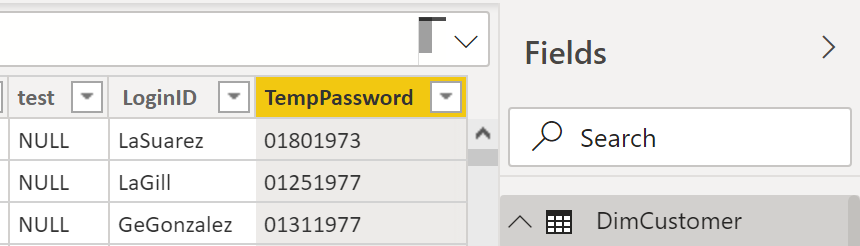


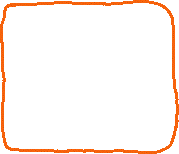
Type “TempPassword” as the new column name, followed by the DAX formula below. Moreover, click on the check mark icon to execute the equation.



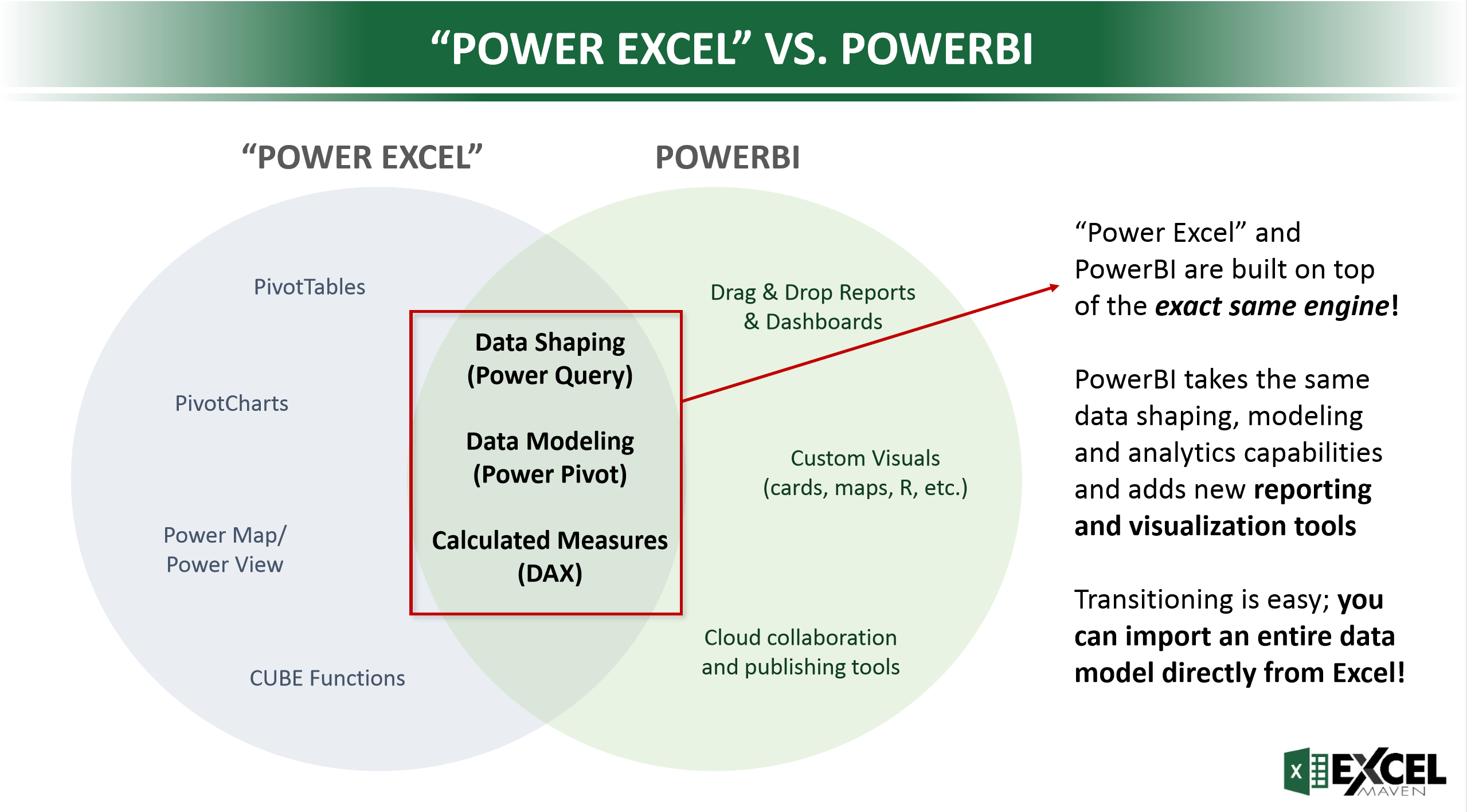


Power BI will work on adding the new column by executing the formula row-by-row. Please verify that the computations are correct. Your first three lines should look like the snippet that follows:





* Please do not forget to save your work regularly.
* As you can see, the use of DAX in Power BI is no different from what you have learned in Power Excel, which was taught in the first half of this term. Why is that? Because DAX engine is the same for both products as illustrated in the diagram below.

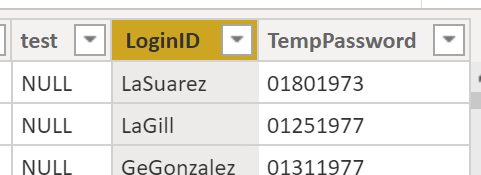


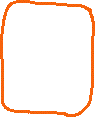


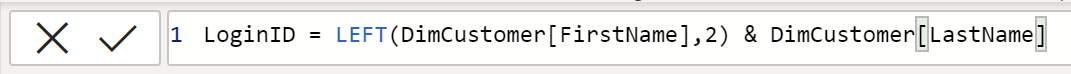
#### Nested function

Microsoft provides extensive documentation of all the [**DAX functions**](https://docs.microsoft.com/en-us/dax/dax-function-reference). For our calculated columns, we will be mostly looking at [**DAX text functions**](https://docs.microsoft.com/en-us/dax/text-functions-dax). Here, you will use the LOWER() [**function**](https://docs.microsoft.com/en-us/dax/lower-function-dax) which converts text to all lower case. We want to use LOWER() to make LoginID fully lowercase, as is standard for usernames. To accomplish this, in this exercise, you will complete this task through the use of nested functions. A nested function is a function within a function like FUNCTION1(FUNCTION2(columnname)).

* In the *Data* view, select LoginID from DimCustomer. You should see the DAX formula in the top pane. Click on the LoginID header, and note how the formula field displays the current equation.







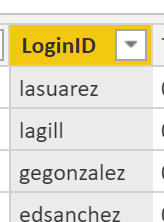


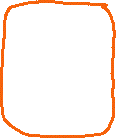
* Edit the formula so that it uses the LOWER() function to make LoginID fully lowercase. Type the changes below and click the check mark icon.





Note how the user IDs below are all in lower case.

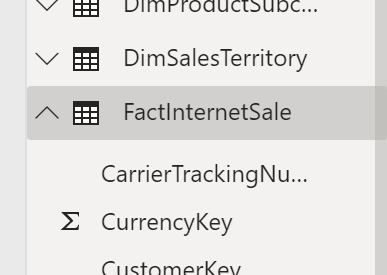




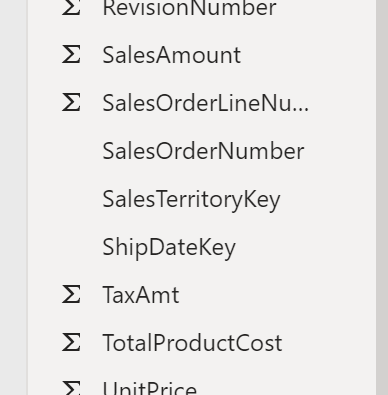
#### Profit column

Along with DAX functions, there are [**DAX operators**](https://docs.microsoft.com/en-us/dax/dax-operator-reference). These operators include arithmetic operations, like + (adding), - (subtracting), and \* (multiplication). Let us try it out!

In the FactInternetSale table, we would like to create a column to see how much profit we gained from each order. To calculate profit, we need to subtract the cost of the product and tax paid from the sales amount. Upon looking at the FactInternetSale table, we see that there are columns for TotalProductCost, TaxAmt, and SalesAmount.

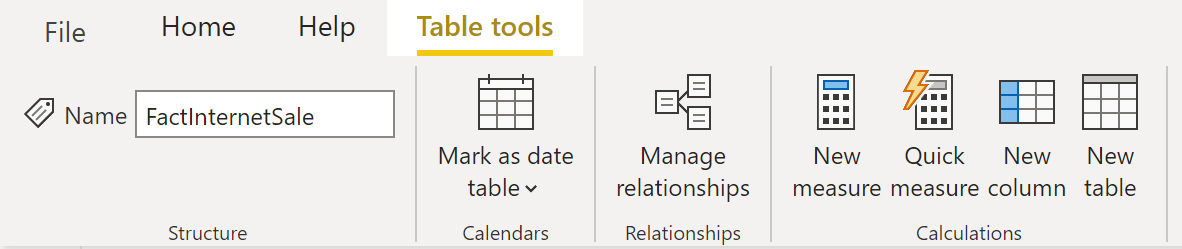






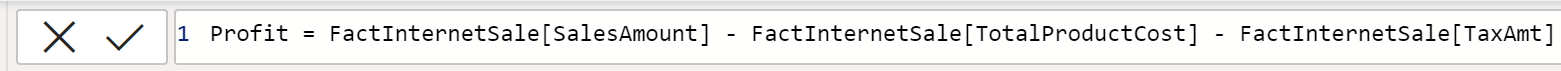


* Create a new column in the FactInternetSale called Profit. While the FactInternetSale table is currently open, click New column.

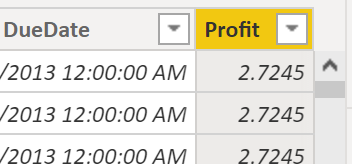




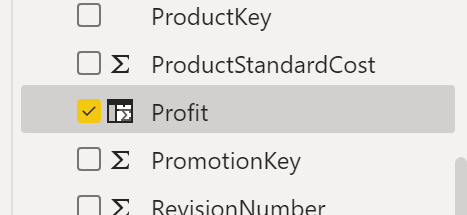
* Type the DAX formula below, and press on the check mark icon.



* A new profit column is now added to the FactInternetSale table.

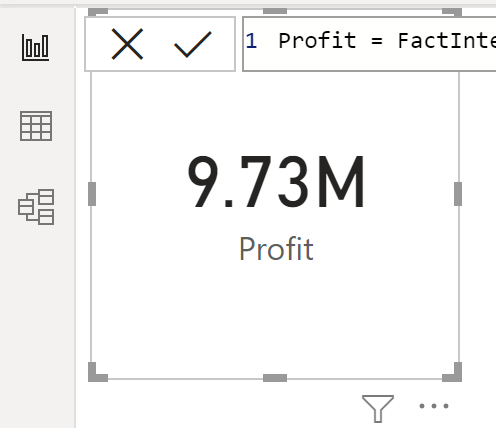


* In the *Report* view, create a card that shows the *sum* of Profit. You can do this by switching to the Report view > click Card visualization widget > click profit field





* Your dashboard should look like the graphics below.



* The point of this short exercise is to make a connection between the DAX technology that you already gained from the first five weeks of this quarter. Those libraries still apply to Power BI—no difference!
* This assignment concludes the Data Visualization class. Thank you for your diligence and passion as we reviewed all aspects of the top business intelligence tool offered by Microsoft. Please dedicate the rest of the class towards working with your team to complete your final project as well as your individual course collage.
* Please reposition your dashboard so that it would show all the encircled items below. Submit your PNG image to Canvas.  Thank you for a great quarter!

