# Lab Data modeling and DAX

## **SCENARIO**

VanArsdel is a company that manufactures and sells sporting goods. The company has offices in the United States (US) and several other countries. Its sales comprise of US sales and International sales. VanArsdel’s sales come from its owned manufactured products, as well as other manufacturers’ products.

You have successfully brought the US sales data from the Access database and the International sales data from a collection of CSV files to Power BI Desktop. Before you can start analyzing your data, you need to manage the table relationships within your data model and create new ones if necessary. To do so, you might need to create calculated columns or calculated tables for the relationships to be based on.

In this lab, you will create calculated columns, calculated tables, and create table relationships in your data model based on the calculated columns and tables you created. In addition, you will write several DAX expressions to create measures to be used to analyze VanArsdel’s sales data.

Before starting this lab, you should review the module about Data Modelling and DAX in this course.

## **WHAT YOU’LL NEED**

* A computer with the latest version of Power BI Desktop installed on it.
* The following Power BI Desktop file: The “Lab Data modelling and DAX - for students.pbix” file

## **Exercise 1: Manage Table Relationships**

Power BI Desktop has automatically detected and created table relationships. So the first step is to ensure all the relationships are properly created, and if not, create them yourselves.

1. Start with the "Lab Data modelling and DAX - for students.pbix" file.
2. Open the **Relationship** view.
3. Ensure that there is a many to one relationship with both cross directional filtering from the ***ProductID***column on the **Sales** table to the ***ProductID***column on the **Products**table. If not, create the relationship by dragging the ***ProductID*** column on the **Sales** table to the ***ProductID*** column on the **Products** table. If there is no relationship, create it.
4. Ensure that there is a many to one relationship with both cross directional filtering from the ***ManufacturerID***column on the **Products**table to the ***ManufacturerID***column on the **Manufacturers**table. If not, create the relationship. If there is no relationship, create it.
5. Ensure that there is a many to one relationship with both cross directional filtering from the ***Date*** column on the **Sales** table to the ***Date*** column on the **Date** table. If not, create the relationship. If there is no relationship, create it.

Now you want to create a relationship between the **Sales** table and the **Locations** table. First, you merge the ***Country*** and ***Zip*** columns in both **Sales** and **Locations** table as a new column, ***CountryZip***. Then, you create a relationship on the ***CountryZip***column for both tables.

1. Edit the **Locations**table in the **Data** view.
2. Add a new column named ***CountryZip***by concatenating the value from the **Country** column, a comma and a space character, and the value from the ***Zip*** column. (Hint: the calculated column formula look as follows: CountryZip = Locations[Country] & ", " & Locations[Zip])
3. Edit the **Sales**table in the **Data** view.
4. Add a new column named ***CountryZip***by concatenating the value from the ***Country Name*** column, a comma and a space character, and the value from the ***Zip*** column. (Hint: the calculated column formula look as follows: CountryZip = Sales[Country Name] & ", " & Sales[Zip])
5. Open the **Relationship** view.
6. Link the newly created ***CountryZip***column on the **Sales** table to the newly created ***CountryZip***column on the **Locations** table.

### **Lab Questions**

Review the relationship that you have just created on the *CountryZip* columns. (Hint: Double-click on the newly created relationship).

* 1. What is the cardinality of the relationship? (underline the answer)

Many to One

One to One

One to Many

Many to Many

Other

* 1. What is the Cross filter direction of the relationship by default? (underline the answer)

Single

Double

Both

Multiple

Other

## **Exercise 2: Creating measures tables**

Before creating any new measures, let us create a table for measures only:

1. Click on Home tab. Choose option Enter data.
2. Create table **\_\_MEASURES\_\_** as in the screenshot below. Leave only default column *Column 1***.**

Graphical user interface, text, application, Word

Description automatically generated

1. Add at least one measure.
2. Go to the Fieldspanel. Right-click the *Column 1***.** Click Delete from model.

Graphical user interface, application

Description automatically generated Graphical user interface, application

Description automatically generated

1. The sign of table should become like that: 
2. From this moment all measures should be written to this table.
3. After that go to Model View**.** Choose any measure in Propertiestab enter the name for the new folder in Display Folderfield.   
   You should keep all your measures in logical folders.

Expected outcome: (the result should look similar, folders can be different). Separate measures table created and folders inside it.

Folders should exist. The specific names do not matter.

Diagram

Description automatically generated

**Exercise 3. Comparing sales between different periods**

You want to know how much sales (revenue) in total the VanArsdel has and to compare this with the figure from the same period last year. You need to create several calculated measures to help with this comparison. To do so, in either the **Report** view or the **Data** view, right-click the **Sales** table, click **New Measure**, and type in the corresponding DAX formulas for the measure you want to create. This will create the measures with the **Home Table** properties set to the **Sales** table.

Specifically, you will create the following measures:

* ***Total Sales***: calculates the total sales. Format this measure ***Currency***. (Hint: Check out the **SUM** function).
* ***Sales LY***: calculates last year sales. Format this measure as **Currency**. (Hint: You might find the **CALCULATE** and **SAMEPERIODLASTYEAR** function useful).
* **Sales Var**: calculates sales difference between this year and last year sales. Format this measure as **Currency**. (Hint: This is simply the difference between ***Total Sales*** and ***Sales LY***).
* ***Sales Var %:*** calculates sales difference between this year and last year sales in percentage. Format this measure as **Percentage**. (Hint: This is simply the percentage of ***Sales Var*** from ***Sales LY***. You might find the **DIVIDE** function useful).

### **Lab Questions**

Answer the following questions using the measures you created. (Do NOT include currency symbols or thousands separators). The fastest way to do this is to drag the measures you created to the Report view and format them as a table visualization. Do not use any level of filtering to answer the questions.

1. What is the figure for the *Total Sales* measure? (to two decimal places)
2. What is the figure for the *Sales Var %* measure? (to two decimal places) %

**Exercise 4: Year to Date**

Year-to-date (YTD) is a period starting from the beginning of the current year and continuing up to the present date. You want to calculate the YTD sales and compare this with the figure from the same period last year. Specifically, you will create the following measures:

* ***YTD Sales***: calculates the YTD sales. Format this measure as **Currency**. (Hint: Check out the **TOTALYTD** function).
* ***YTD LY Sales***: calculates last year YTD sales. Format this measure as **Currency**. (Hint: You might find the **CALCULATE** and **SAMEPERIODLASTYEAR** function useful).
* ***YTD Sales Var***: calculates sales variance between this year and last year YTD sales. Format this measure as **Currency**. (Hint: This is simply the difference between ***YTD Sales*** and ***YTD LY Sales***).
* ***YTD Sales Var %***: calculates sales variance between this year and last year YTD sales in percentage. Format this measure as **Percentage**. (Hint: This is simply the percentage of ***YTD Sales Var*** from ***YTD LY Sales***. You might find the **DIVIDE** function useful).

### **Lab Questions**

Answer the following questions using the measures you created. (Do NOT include currency symbols or thousands separators). The fastest way to do this is to drag the measures you created to the Report view and format them as a table visualization. Do not use any level of filtering to answer the questions.

1. What is the figure for the *YTD LY Sales* measure? (to two decimal places) 
2. What is the figure for the *YTD Sales Var %* measure? (enter the **absolute** value, to two decimal places)  %

## **Exercise 5: Market Share**

VanArsdel’s sales comprise of products manufactured by VanArsdel and other companies. You want to know how much of these sales are VanArsdel’s own manufactured products. You decide to show this share in numbers and %. Specifically, you will create the following measures:

* ***VanArsdel Total Sales***: calculates sales where the products manufacturer is VanArsdel. Format this measure as **Currency**. (Hint: Use the **CALCULATE** function and filter by Manufacturer).
* ***VanArsdel % Sales Market Share***: calculates the percentage of sales of VanArsdel manufactured products from the total sales. Format this measure as **Percentage**.

### **Lab Questions**

Answer the following questions using the measures you created. (Do NOT include currency symbols or thousands separators). The fastest way to do this is to drag the measures you created to the Report view and format them as a table visualization. Do not use any level of filtering to answer the questions.

1. What is the figure for the *VanArsdel* *Total Sales* measure? (to two decimal places)
2. What is the figure for the *VanArsdel* *% Sales Market Share* measure? (to two decimal places) %

**Exercise 6: Optimize the Data Model**

Now that you have the table relationships defined and the measures created, you want to optimize the data model before you create the visualizations.

1. Open the **Data** view.
2. Ensure both the **International Sales** and **Country Population** table are hidden from the report view.
3. Hide the following fields on the **Date** table from the report view.
   * ***MonthNo***
   * ***MonthID***
   * ***Month***
4. Sort the ***MonthName*** column by the **MonthNo** column.
5. Hide the ***CountryZip***field on the **Locations**table from the report view.
6. Hide the ***ManufacturerID*** field on the **Manufacturers** table from the report view.
7. Hide the following fields on the **Products**table from the report view.
   * ***ProductID***
   * ***ManufacturerID***
   * ***Manufacturer***
8. Hide the following fields on the **Sales**table from the report view.

* ***Date***
* ***Zip***
* ***Units***
* ***Revenue***
* ***Country Name***
* ***CountryZip***

**Exercise 7: Revenue per unit calculation.**

Market changes from time to time and it reflects on unit prices that VanArsdel manufactures and sells.

Prices fluctuations can be mitigated if we calculate average price per each product per year. Afterwards price change can be visualized.

1. Go to **Sales** table
2. Create column called *Price***.** To make it divide revenue by units with the help of DIVIDEfunction. Now you can see the prices of all products for all days but it is will be hard to find any trend with such calculation.
3. Create new column called *Year*using YEAR()function.
4. Next step is calculating total revenue per each product for every year. To do this create new column *Revenue\_per\_product\_year***.** The formula should include: CALCULATE/ALLEXCEPTwith *ProductID*and *Year***;**

In Data view you can see that *Revenue\_per\_product\_year*is the same for every product in one year.

1. Calculate *Units\_per\_product\_year*the same way.
2. Create new column *Price\_per\_product\_year*with the help of DIVIDEfunction
3. Go to Data View.

What year was the highest price per product&year for productid = 1000? 

And what is the price? 

**Exercise 8: TOP/BOTTOM products.**

VanArsdel top management wants to understand the products with highest sales and the products with lowest sales. Top and bottom products can change over time. That is why calculation should be dynamic.

1. Go to **Measures** table
2. Create measure *Product Rank DESC*(Hint: it should use operator RANKX, *ProductID* column from **Sales** table, DESC and DENSE values inside operator).  
   The result should be the same as in screenshot.

Table

Description automatically generated

1. Create similar measure *Product Rank ASC***.** It should return higher rank for products with lowest sales.
2. Write down the number of products whose *Product Rank ASC* is <= 10. 

Another way of achieving similar result.

1. Use *Revenue\_per\_product\_year*field to create bins. In Bin Typechoose Number of binsand type 10. Right now you divided all products on 10 segments according to its’ sales for the whole history.
2. Create new calculated column as dense rank and call it *Custom\_product\_rank***.** Use *Revenue\_per\_product\_year* for rank calculation.
3. Create bar chart where you calculate the number of products per each *Custom\_product\_rank***.**

Write down how many unique products is there with *Custom\_product\_rank = 9***.** 

## **Exercise 9. Using SWITCH operator**

Sometimes stakeholders want to see same visualizations with different values. The best way to solve this issue is separate slicer for measures.

1. On Home tabchoose Enter data**.**
2. Create table **CHOOSE\_MEASURE** with the following values: “Sales”, “Units”.

Table

Description automatically generated

1. Create new measure *Total Units* = SUM(Units)
2. Write the new measure *Chosen\_measure*that returns *Total Sales*if we choose “Sales” value in slicer and *Total Units*if “Units” value was chosen. The measure should include SWITCHand SELECTEDVALUE
3. Create a simple visualization with a slicer “Measure” at the top (slicer should be single-valued) and horizontal bar chart per country at the bottom. Values in visualization should change if you choose the different value in “Measure”.
4. Attach the screenshot of visualization here

A blue line with white text

Description automatically generated

## **Exercise 10. Comparing values for custom periods**

Usually, it is enough for business users to compare values for well-known periods such YTD/MTD/Same period for last year/Previous period. However, there are cases when it is much more useful to compare calculations for different periods. The best way to do this is 2 calendar tables.

1. Go to Modelling – New Table**.** Write an expression: Date\_Previous = 'Date' **.** This way you should create a new table that is full copy of **Date** table.
2. Go to Model View**.** Create a new 1-to-1 inactive relationship between **P-** and **Date** tables based on *Date* column.
3. Create new measure *Total Sales, previous period*. It should calculate *Total Sales*for selected period. As we have relationship between 2 calendar tables, we should get rid of all filters in **Date** table to get the result for previous periods. To write the correct formula you should activate the relationship when it is need with the help of USERELATIONSHIPand delete all **Date** filters with the of REMOVEFILTERS**.** Use CALCULATE function for this.
4. Create new measure called *Total Sales, diff PY*that is just the difference between *Total Sales*and *Total Sales, previous period.*
5. In Report View create new table and 2 slicers. 1st slicer is year from **Date** table. 2nd slicer is year from **Date\_previous** table. Table should include values for *Total Sales***;** *Total Sales, previous period***;** *Total Sales, diff PY.*There is example in the screenshot.

Table

Description automatically generated

1. Write down the result of *Total Sales, diff PY*if 1st slicer values include 2000, 2005,2012 years and 2nd slicer includes 2001,2002,2003,2009. 