Writer:	I&D Microsoft	**Initials entity:**
Date creation: 24/04/2018		**Last upgrade:** 25/04/2018

POWER BI LAB DOCUMENT

DAY 6- LAB 1

Version	Author	Comment	Reviewed By	Date
V 1.0	I&D Microsoft	Initial draft	Moupiya Das	

Pre-requisites

Installed and working Power BI Desktop setup.

Environment Setup

1. Installing Power BI Desktop

Power BI Desktop lets you create a collection of queries, data connections, and reports that can easily be shared with others. Power BI Desktop integrates proven Microsoft technologies – the powerful Query engine, data modeling, and visualizations – and works seamlessly with the online Power BI service. You will need to download and install Power BI desktop to perform the labs in this course.

1.1 Which version to install?

- If you have a 32 bit machine, you need to install the 32bit Power BI Desktop.
- If you have a 32 bit Office installed (regardless of your machine), you need to install the 32 bit Power BI Desktop.
- Otherwise, you can install the 64 bit PBI Desktop.

1.2 Minimum requirements

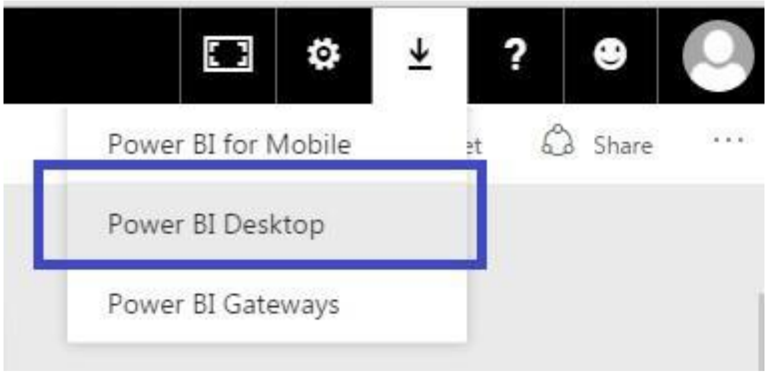
- Windows 7 / Windows Server 2008 R2, or later
- .NET 4.5
- Internet Explorer 9 or later

1.3 Download and install Power BI Desktop

You can download and install the latest version of Power BI Desktop in two ways,

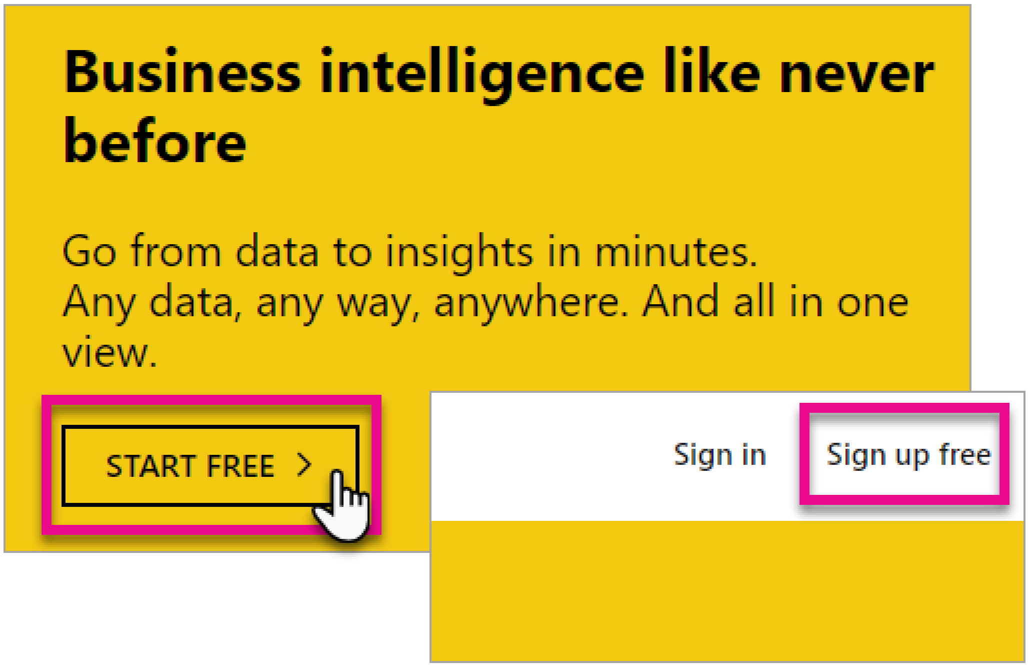
1. Directly from Microsoft link, http://go.microsoft.com/fwlink/?LinkID=521662
Note: To select which version to download, go to https://www.microsoft.com/en-us/download/details.aspx?id=45331 and select appropriate version.

2. Or, from Power BI Service site, https://app.powerbi.com/, i.e., in Power BI, click the Downloads > **Power BI Desktop**.



2. Signing Up for Power BI Service

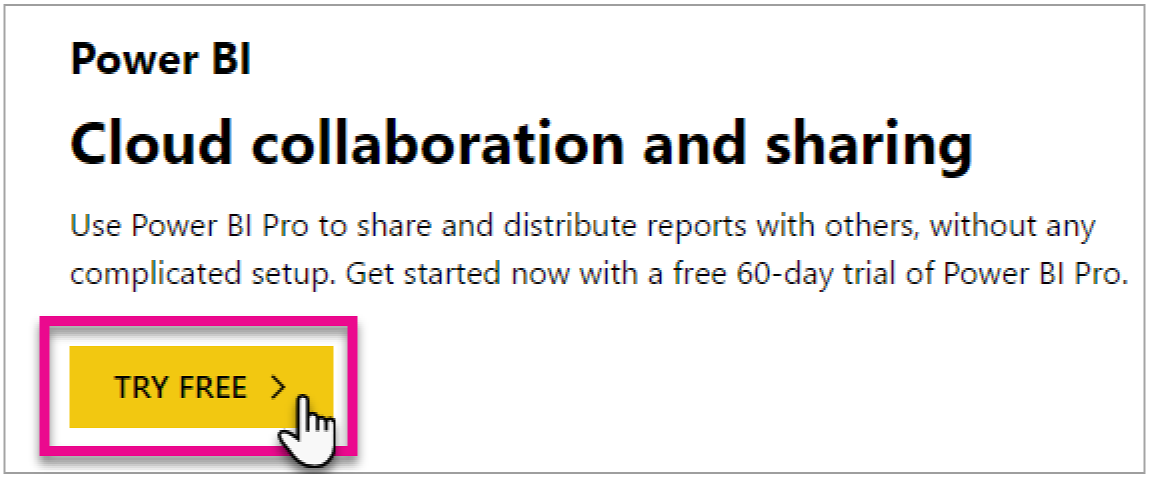
1. Browse to powerbi.com.
2. Select **Start Free** or **Sign up free**.



Business intelligence like never before

Go from data to insights in minutes.
Any data, any way, anywhere. And all in one view.

3. On the get started page, select **Try Free >** under Power BI.

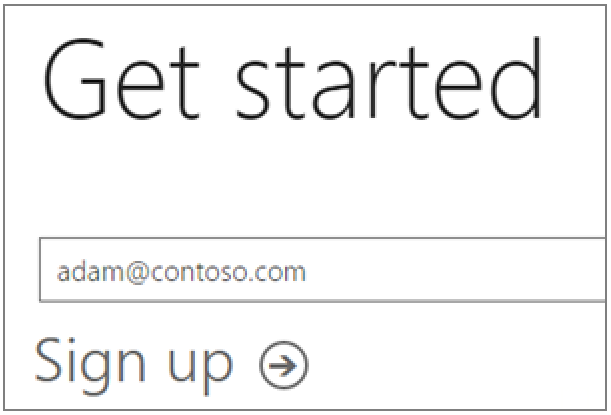


Power BI

Cloud collaboration and sharing

Use Power BI Pro to share and distribute reports with others, without any complicated setup. Get started now with a free 60-day trial of Power BI Pro.

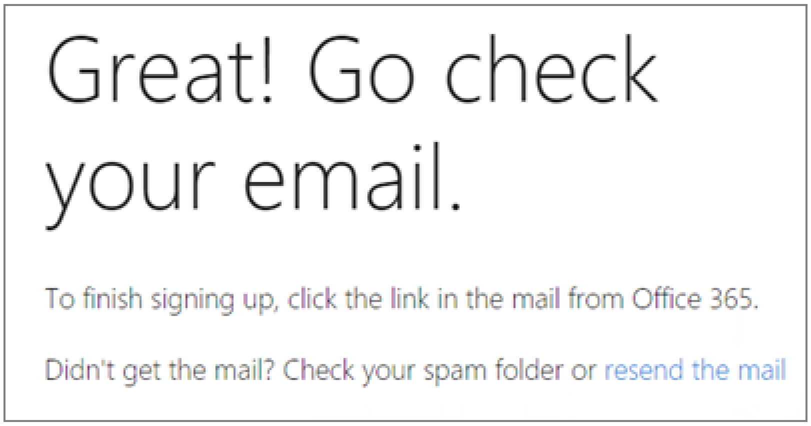
4. Enter the email address you are signing up with, and then select **Sign up**. Be sure your email address is allowed for sign up. For more information about what email address you can use, see What email address can be used with Power BI.



Get started

adam@contoso.com

5. You will get a message indicating to check your email.



Great! Go check your email.

To finish signing up, click the link in the mail from Office 365.

Didn't get the mail? Check your spam folder or resend the mail

6. Select the link within the email to verify your email address. This will bring you back into the sign up flow. You may need to supply some additional information about yourself.
7. You will then be taken to https://app.powerbi.com and you can begin using Power BI as a free user.

Lab Overview

This lab comprises of four sections:

1. In the first section, you will create role based security based on territory region.

2. In the second section, you will basic DAX measures.

3. In the third section, you will create few measures and understand how they are different from calculated columns by using Time Intelligence function.

4. In the fourth section you will understand how calculations work in DAX by means of context and context transition.

Table of Contents

Table of Contents 7

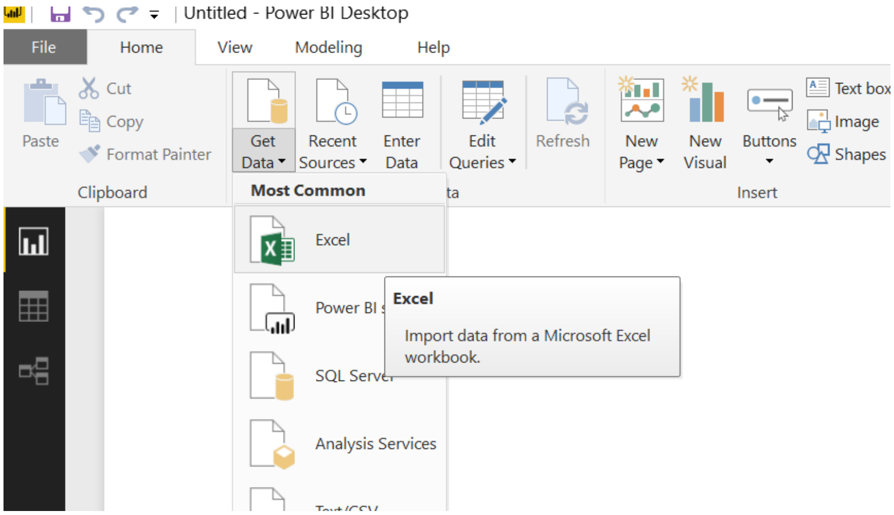
1. RLS – Role Based Security Lab 8
2. Creating Measures Lab 17
3. Time Intelligence & Calculate column vs Measures Lab 19
4. Understanding DAX computation logic, Context Transition Lab 21

1. RLS – Role Based Security Lab

Case Scenario

This case requires sales managers to view data from their respective allocated region. To complete this requirement, implement Role based security on the data by creating roles based on region and provide membership to users for their respective region. That way, members belonging to their region will see only their data and not the data related to another region. Now let us implement this using the data from Adventure works sample excel file (You should be able to find this in the file: ***AdventureWorksDatabase.xlsx***).

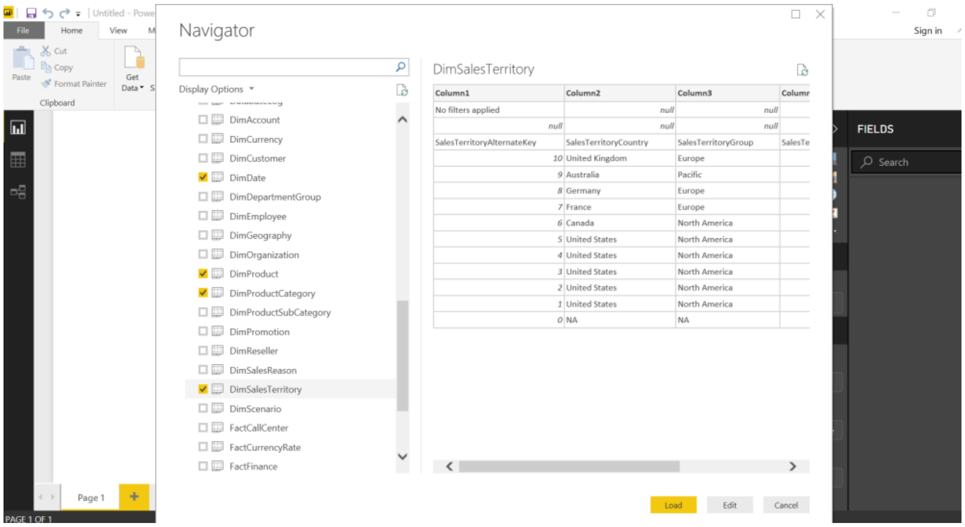
1. Open the **Power BI Desktop**. You can try to sign if it asks for provided you have access to **Power BI Service** as demonstrated in earlier labs. Then, click on Get Data → Excel and browse the excel file for *AdventureWorksDatabase.xlsx*.



Excel

Import data from a Microsoft Excel workbook.

2. Select DimDate, DimProduct, DimProductCategory, DimSalesTerritory and FactInternetSales. Click **Load** to load the data. You must see the below screen in this process.

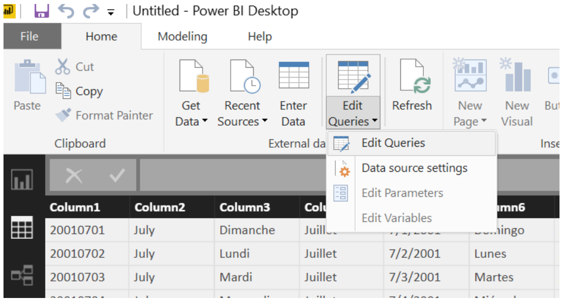


- ☐ DimAccount
- ☐ DimCurrency
- ☐ DimCustomer
- ☑ DimDate
- ☐ DimDepartmentGroup
- ☐ DimEmployee
- ☐ DimGeography
- ☐ DimOrganization
- ☑ DimProduct
- ☑ DimProductCategory
- ☐ DimProductSubCategory
- ☐ DimPromotion
- ☐ DimReseller
- ☐ DimSalesReason
- ☑ DimSalesTerritory
- ☐ DimScenario
- ☐ FactCallCenter
- ☐ FactCurrencyRate
- ☐ FactFinance

DimSalesTerritory

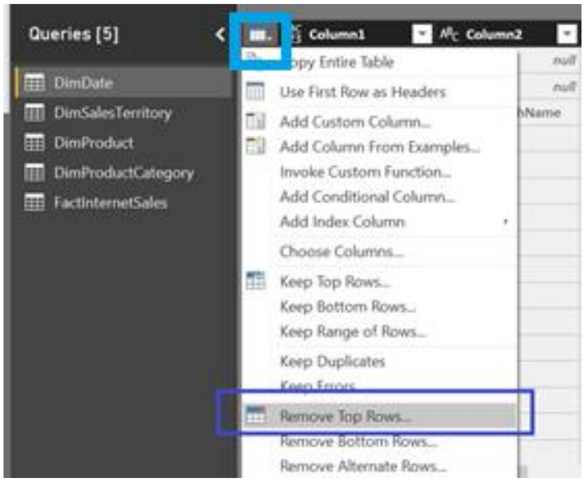
Column1	Column2	Column3	Columr
No filters applied	null	null	
null	null	null	
SalesTerritoryAlternateKey	SalesTerritoryCountry	SalesTerritoryGroup	SalesTe
10	United Kingdom	Europe	
9	Australia	Pacific	
8	Germany	Europe	
7	France	Europe	
6	Canada	North America	
5	United States	North America	
4	United States	North America	
3	United States	North America	
2	United States	North America	
1	United States	North America	
0	NA	NA	

3. Once data is loaded, click Home → Edit Queries → Edit Queries

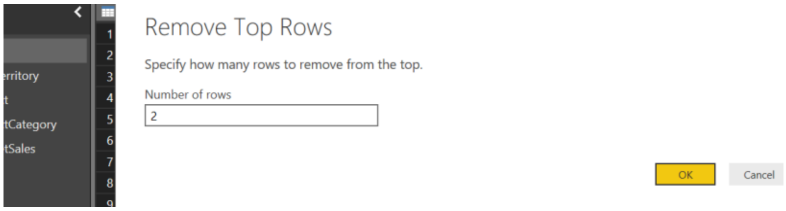


Column1	Column2	Column3	Col		mn6
20010701	July	Dimanche	Juill		ingo
20010702	July	Lundi	Juillet	7/2/2001	Lunes
20010703	July	Mardi	Juillet	7/3/2001	Martes

4. Click the first query (DimDate in the below screenshot). Click the table icon on the right of **Queries** (highlighted in below screenshot) → click on **Remove Top Rows…** and a window will appear as in the next step.



5. Enter 2 as the **Number of rows** textbox to be removed from the top. This is done because the first 2 rows in the loaded data is junk data and does not need to be a part of the dataset.

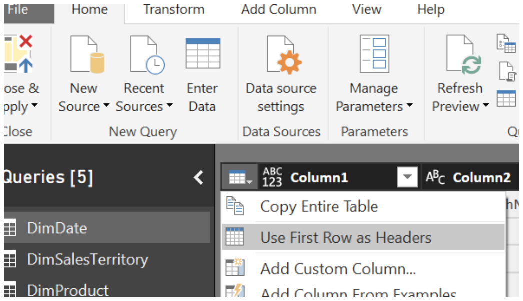


Remove Top Rows

Specify how many rows to remove from the top.

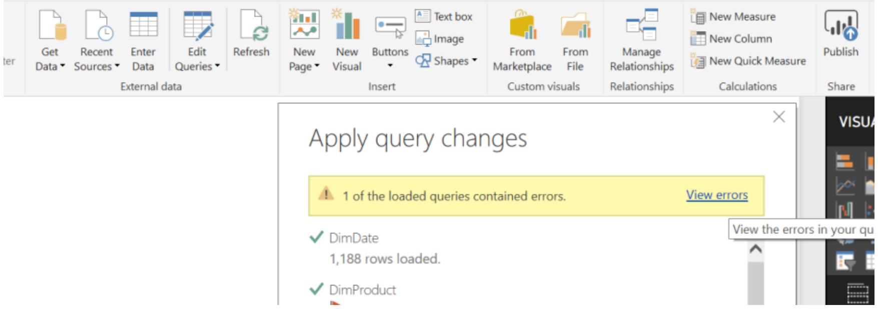
Number of rows: 2

6. Now, the first record shown in the data is a header, so let’s promote that as header record. Click on the table icon again, and click **Use First Row as Headers**.



7. Perform the same operation for other Queries like DimSalesTerritory, DimProduct and so on.

8. Load the data after this operation. You won’t face any error now; however, if you face errors try to resolve by considering the query.



Apply query changes

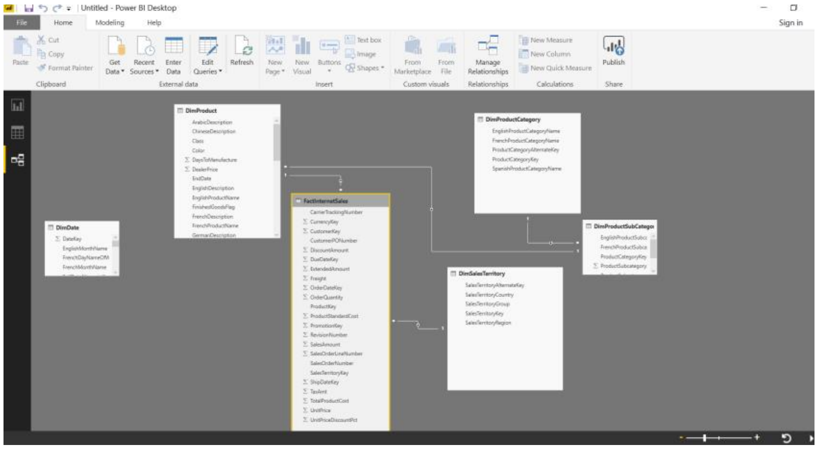
1 of the loaded queries contained errors. View errors

View the errors in your qu

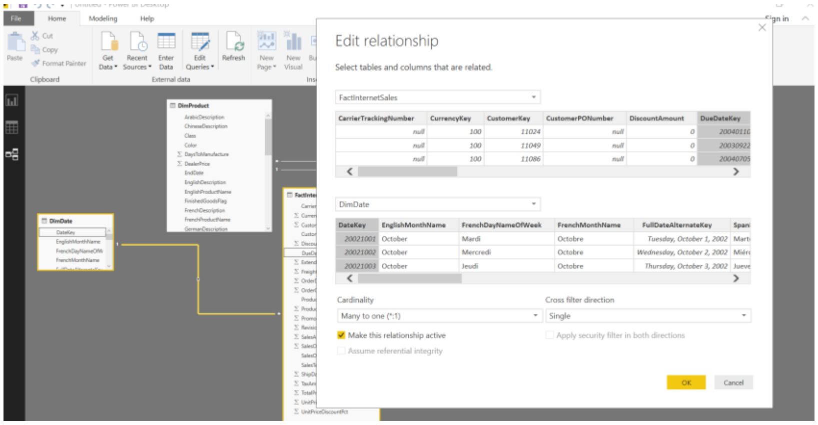
✓ DimDate
1,188 rows loaded.

✓ DimProduct

9. Check the relationship tab and the data model should now look like:



10. Click on Manage relationship and create a new relationship between DimDate and FactInternetSales as FactInternetSales.OrderDateKey = DimDate.DateKey. Select Ok to activate this relationship.



Edit relationship

Select tables and columns that are related.

FactInternetSales

CarrierTrackingNumber	CurrencyKey	CustomerKey	CustomerPONumber	DiscountAmount	DueDateKey
null	100	11024	null	0	2004011C
null	100	11049	null	0	20030922
null	100	11086	null	0	20040705

DimDate

DateKey	EnglishMonthName	FrenchDayNameOfWeek	FrenchMonthName	FullDateAlternateKey	Spani
20021001	October	Mardi	Octobre	Tuesday, October 1, 2002	Mart
20021002	October	Mercredi	Octobre	Wednesday, October 2, 2002	Miérc
20021003	October	Jeudi	Octobre	Thursday, October 3, 2002	Jueve

Cardinality: Many to one (*:1)

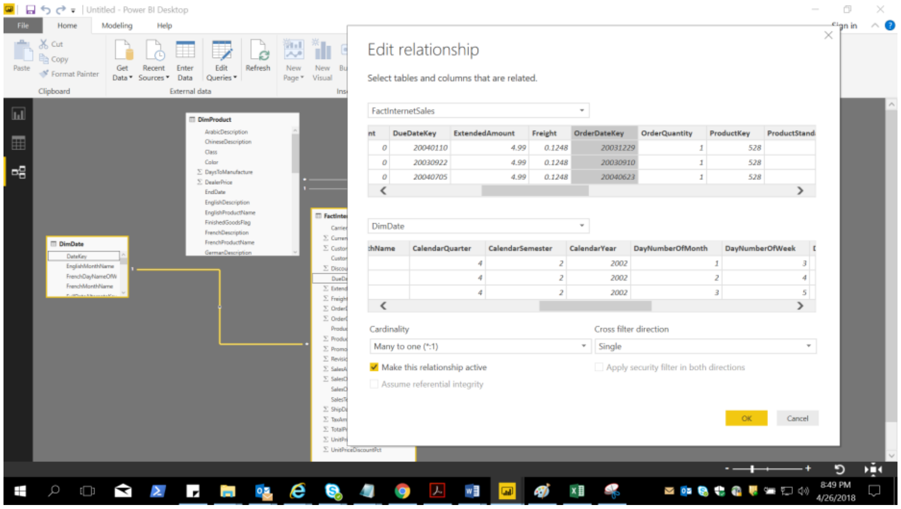
Cross filter direction: Single

☑ Make this relationship active

☐ Apply security filter in both directions

☐ Assume referential integrity

The relationship under the relationships tab should now look like:



Edit relationship

Select tables and columns that are related.

FactInternetSales

nt	DueDateKey	ExtendedAmount	Freight	OrderDateKey	OrderQuantity	ProductKey	ProductStand
0	20040110	4.99	0.1248	20031229	1	528	
0	20030922	4.99	0.1248	20030910	1	528	
0	20040705	4.99	0.1248	20040623	1	528	

DimDate

thName	CalendarQuarter	CalendarSemester	CalendarYear	DayNumberOfMonth	DayNumberOfWeek
	4	2	2002	1	3
	4	2	2002	2	4
	4	2	2002	3	5

Cardinality: Many to one (*:1)

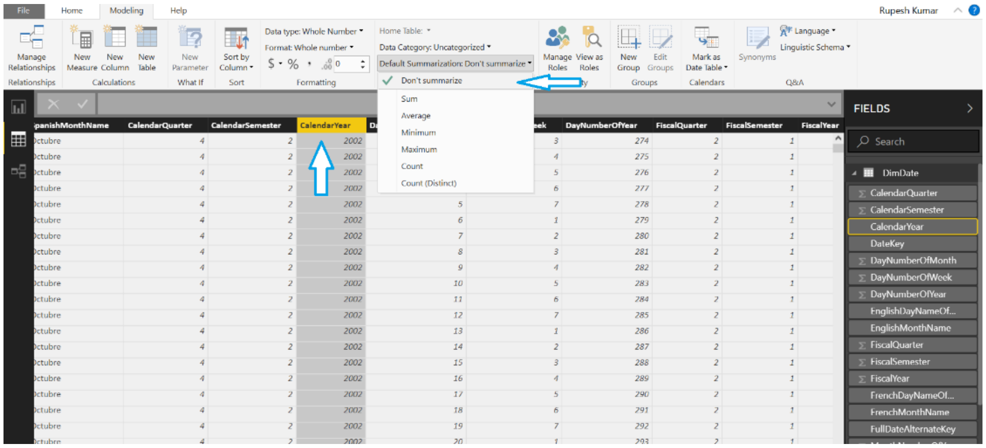
Cross filter direction: Single

☑ Make this relationship active

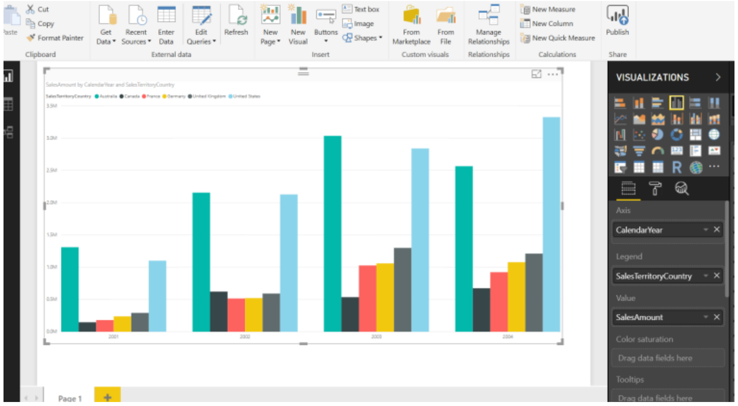
☐ Apply security filter in both directions

☐ Assume referential integrity

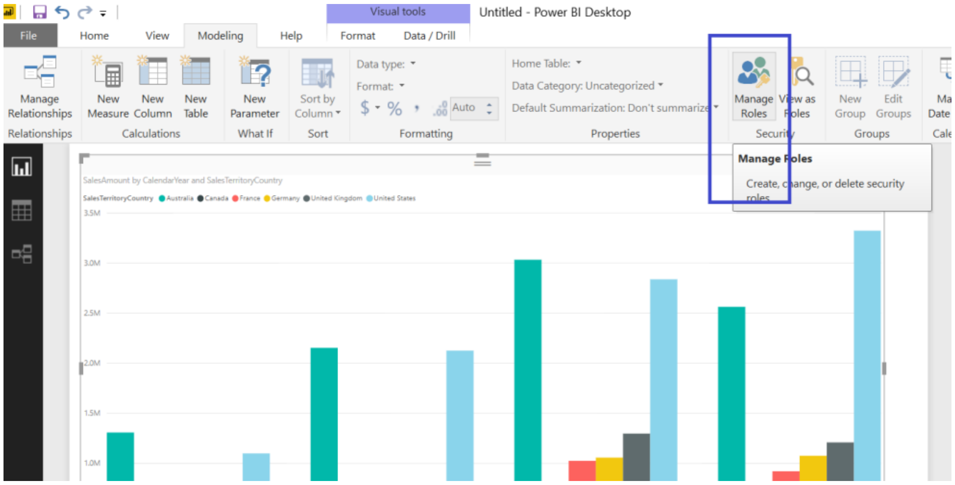
11. Navigate data from different dimensions and fact. Click the DimDate and click on the CalendarYear column. Observe the **Data Type**, **Format**, **Data Categorization** and **Default Summarization** under the **Modeling** tab. Change the **Default Summarization** to Don't Summarize for CalendarYear column.



12. Now, create a **Clustered Column chart** visual with the fields: **DimDate.CalendarYear**, **DimSalesTerritory.SalesTerritoryCountry** and **FactInternetSales.SalesAmount**.

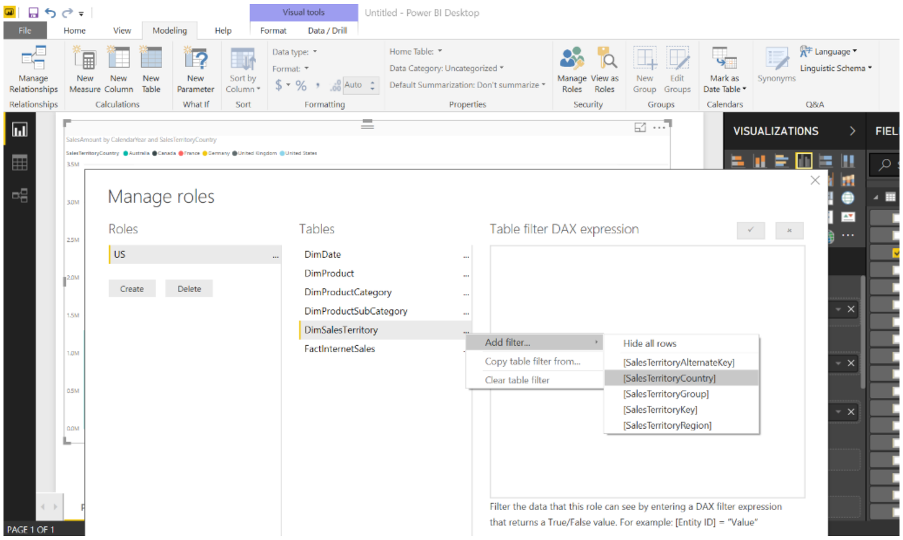


13. After thus, click **Modeling → Manage Roles**

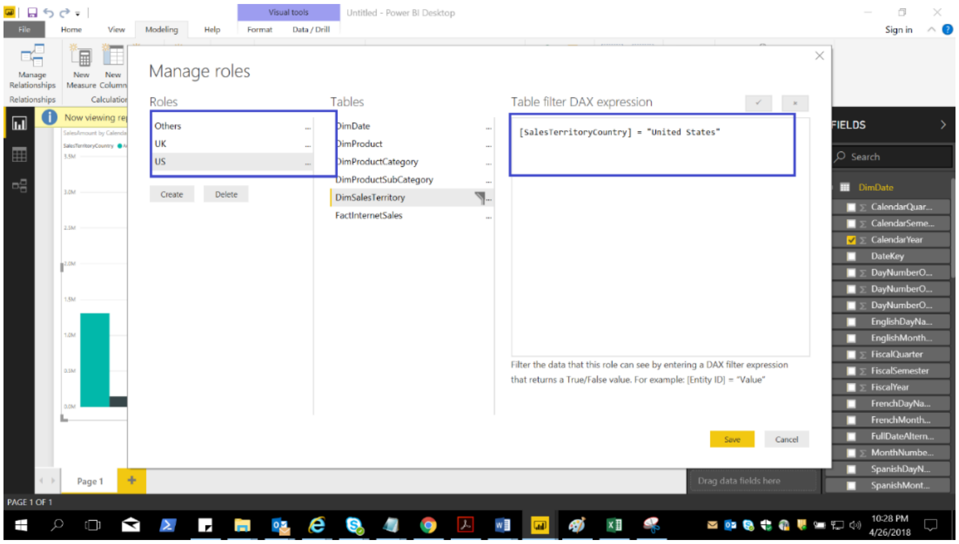


14. A window should now appear like the below. Start creating new roles here by clicking **Create**. Double click on the role names to rename the roles. For the first role, rename it to **US**, then click the 3 ellipses on the right of DimSalesTerritory, hover the mouse to Add filter and click on

[SalesCategoryCountry]. This will populate the DAX expression under the Table filter DAX expression window.



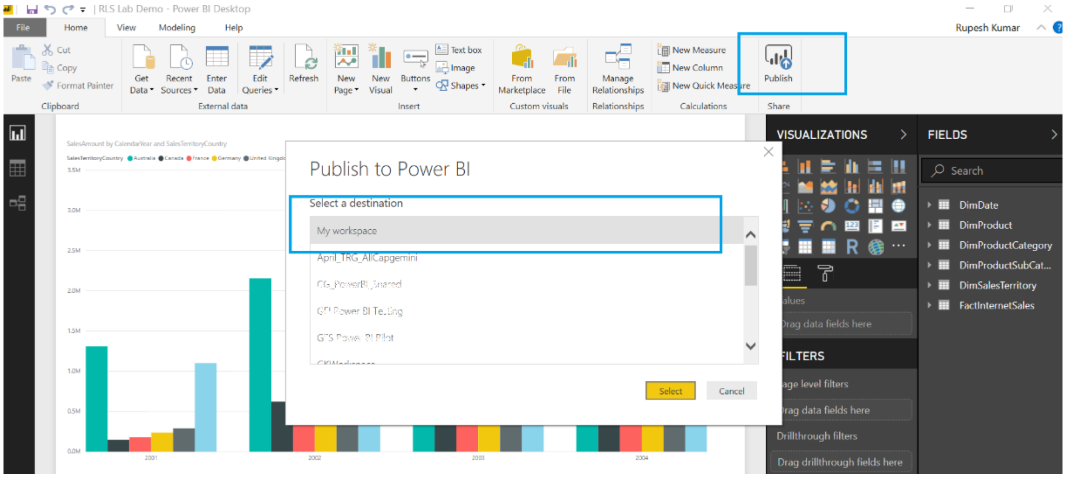
Change the DAX expression so that the value is “**United States**”.


Manage roles
Roles
Others
UK
US
Create
Delete
Tables
DimDate
DimProduct
DimProductCategory
DimProductSubCategory
DimSalesTerritory
FactInternetSales
Table filter DAX expression
Filter the data that this role can see by entering a DAX filter expression that returns a True/False value. For example: [Entity ID] = "Value"
Save
Cancel
FIELDS
DimDate
PAGE 1 OF 1

```
[SalesTerritoryCountry] = "United States"
```

Follow the same rules to create 2 more roles for UK and Others. The **UK** DAX expression should be **[SalesTerritoryCountry] = "United Kingdom"** and **Others** should be **[SalesTerritoryCountry] <> "United States" && [SalesTerritoryCountry] <> "United Kingdom"**

15. Save this PBI file by the name **RLS Lab Demo**.
16. For these next set of tasks make sure you have access to Power BI Service and logged in. Click on the Publish to publish the newly created. Select **My workspace** so that the PBI file gets published to your workspace.

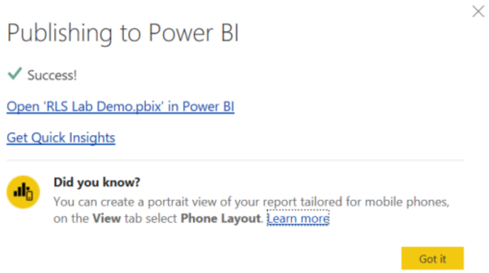


Publish to Power BI

Select a destination

My workspace
April_TRG_AllCapgemini
CG_PowerBI_Shared
GFI Power BI Testing
GTS Power BI Pilot
CKWorkspace

On successful publishing you should get the below message:



Publishing to Power BI

✓ Success!

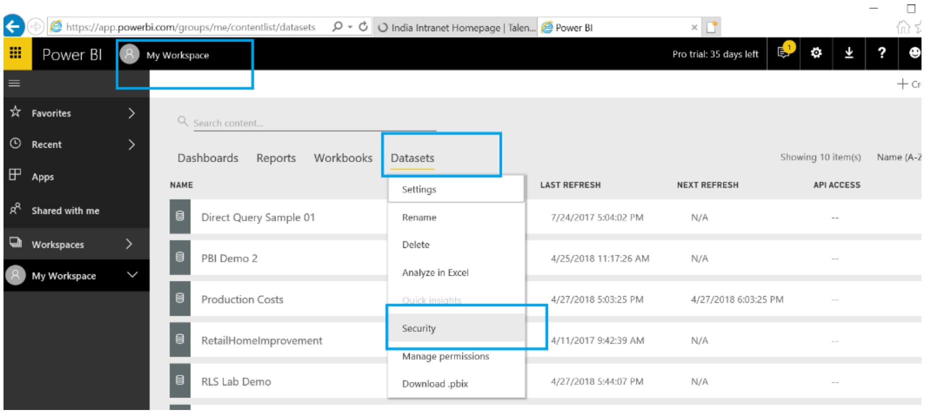
Open 'RLS Lab Demo.pbix' in Power BI

Get Quick Insights

Did you know?

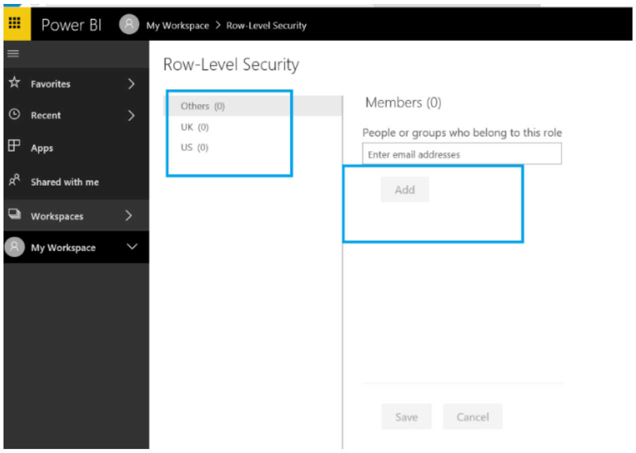
You can create a portrait view of your report tailored for mobile phones, on the **View** tab select **Phone Layout**. Learn more

17. Log on to Power BI portal and navigate to My workspace. Click on the **Datasets** and click on the 3 ellipses under **Actions** and to the right of RLS Lab Demo to get **More Options → Security**.

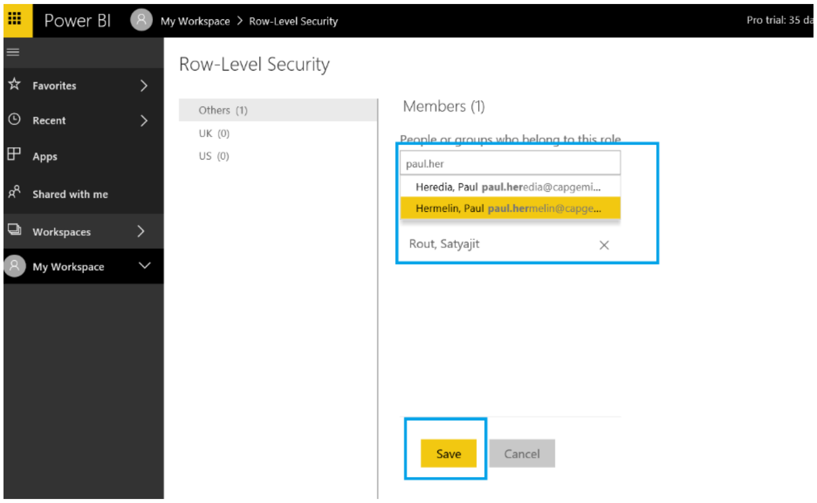


NAME	LAST REFRESH	NEXT REFRESH	API ACCESS
Direct Query Sample 01	7/24/2017 5:04:02 PM	N/A	--
PBI Demo 2	4/25/2018 11:17:26 AM	N/A	--
Production Costs	4/27/2018 5:03:25 PM	4/27/2018 6:03:25 PM	--
RetailHomeImprovement	4/11/2017 9:42:39 AM	N/A	--
RLS Lab Demo	4/27/2018 5:44:07 PM	N/A	--

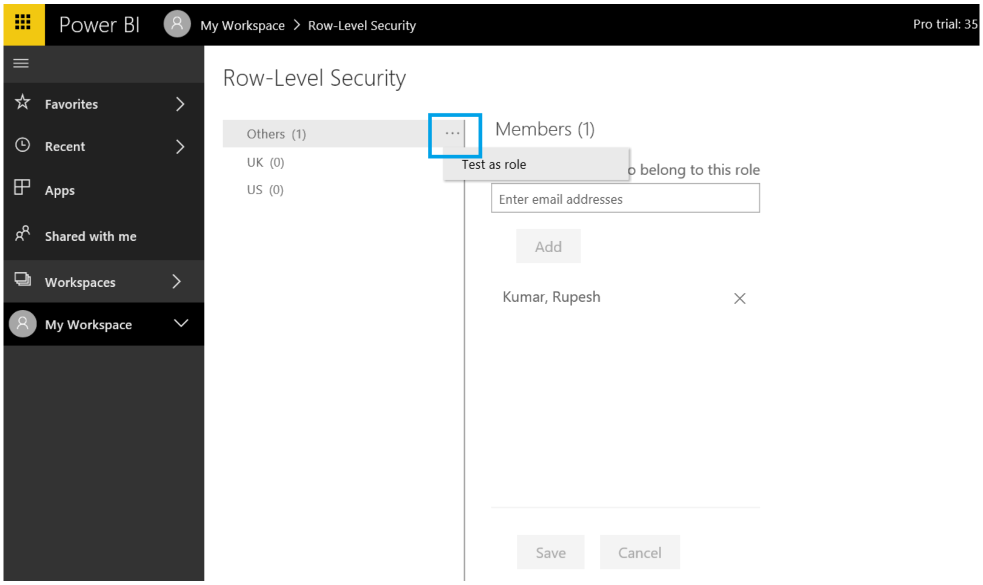
Row-Level Security should now appear like the one below:



Click on a particular role and under **Members**, type an e-mail address to whom the role should be associated. Add a few names to different roles and click **Save** to save the changes.



18. Let is test the roles just created. Click on the 3 ellipses on the right of any role and click the **Test as role**. Observe by clicking different roles and the data that appear in the chart.



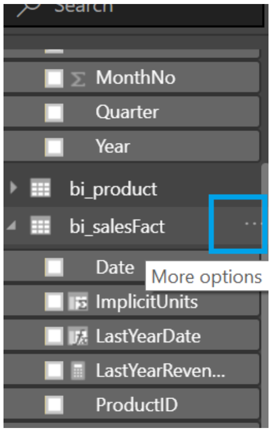
Summary:

This completes the lab session where you have observed how different users under different roles views data belonging to their own roles defined by the DAX filter expression. You can perform interactive testing by asking assigned Users actually log in to Power BI and verify that row level filters are working.

2. Creating Measures Lab

This lab is meant to expose users to creating basic measures and observe how they work.

1. Open the SalesSample.pbix file that is preloaded with the data on PBI Desktop.
2. Create a **Table** visualization with the following attributes: -
   (a) **bi_date** : Year and MonthNo
   (b) **bi_salesFact**: Revenue
3. Click on the ellipse (3 dots) beside **bi_salesFact** and create a New Measure



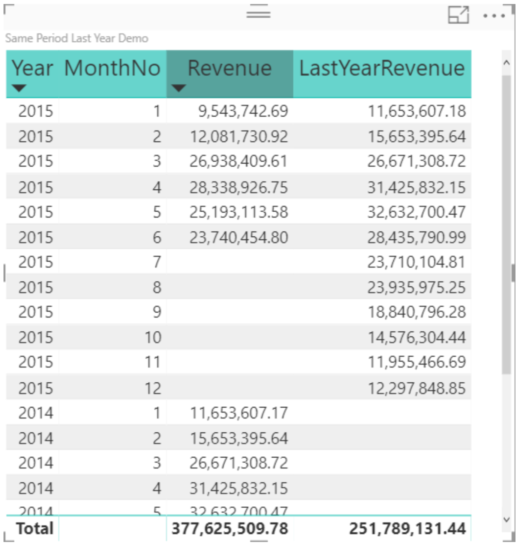
Paste the text of the new measure as:

```
TotalRevenue = SUM(bi_salesFact[Revenue])
```

Next, Create another measure and use the below DAX expression to define it

```
LastYearRevenue = CALCULATE(SUM(bi_salesFact[Revenue]),
SAMEPERIODLASTYEAR(bi_date[Date]))
```

4. Use this measure (TotalRevenue) in the **Table** visual just created and should look like:



Year	MonthNo	Revenue	LastYearRevenue
2015	1	9,543,742.69	11,653,607.18
2015	2	12,081,730.92	15,653,395.64
2015	3	26,938,409.61	26,671,308.72
2015	4	28,338,926.75	31,425,832.15
2015	5	25,193,113.58	32,632,700.47
2015	6	23,740,454.80	28,435,790.99
2015	7		23,710,104.81
2015	8		23,935,975.25
2015	9		18,840,796.28
2015	10		14,576,304.44
2015	11		11,955,466.69
2015	12		12,297,848.85
2014	1	11,653,607.17	
2014	2	15,653,395.64	
2014	3	26,671,308.72	
2014	4	31,425,832.15	
2014	5	32,632,700.47	
Total		**377,625,509.78**	**251,789,131.44**

Observe how the **Last Year Revenue** is calculated based on year and month number. Compare the values of 2014 and 2015 for **Revenue** and **LastYearRevenue**.

5. Now remove **MonthNo** and add **Quarter** instead. Observe how the values change both for **Revenue** and **LastYearRevenue**.

Create one more measure as follows:

```
Variance Per = ([TotalRevenue]-[LastYearRevenue])/[TotalRevenue]
```

Add this to the existing visual.

6. Create a **Card** visual and add **Variance Per** measure just created. Compare this with the values in Table visual just created to understand how variance percentage is dynamically calculated based on the evaluation context. Save and close the Power BI file.

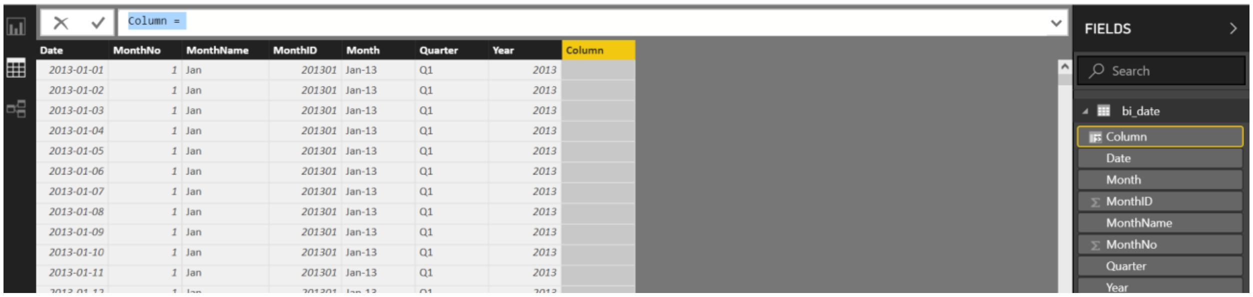
Summary:

This completes the Power BI lab session for basic understanding of measures and how it works.

3. Time Intelligence & Calculate column vs Measures Lab

This lab is targeted in understanding Time intelligence functions and how it works in Power BI. Let us start this session by creating few time related functions.

1. Click on the **Modeling** tab → Select the Data → **New Column** to add a new column.



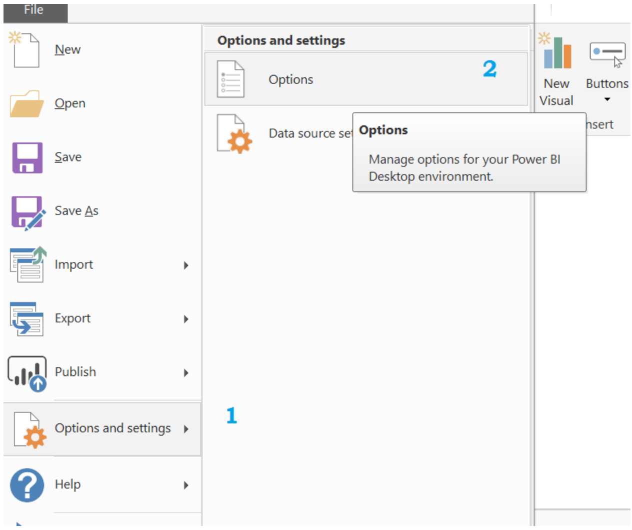
```
Column =
```

Date	MonthNo	MonthName	MonthID	Month	Quarter	Year	Column
2013-01-01	1	Jan	201301	Jan-13	Q1	2013	
2013-01-02	1	Jan	201301	Jan-13	Q1	2013	
2013-01-03	1	Jan	201301	Jan-13	Q1	2013	
2013-01-04	1	Jan	201301	Jan-13	Q1	2013	
2013-01-05	1	Jan	201301	Jan-13	Q1	2013	
2013-01-06	1	Jan	201301	Jan-13	Q1	2013	
2013-01-07	1	Jan	201301	Jan-13	Q1	2013	
2013-01-08	1	Jan	201301	Jan-13	Q1	2013	
2013-01-09	1	Jan	201301	Jan-13	Q1	2013	
2013-01-10	1	Jan	201301	Jan-13	Q1	2013	
2013-01-11	1	Jan	201301	Jan-13	Q1	2013	
2013-01-12	1	Jan	201301	Jan-13	Q1	2013	

2. Edit the DAX query replace with `TimeNow = NOW()`

Observe the time that gets into the new column.

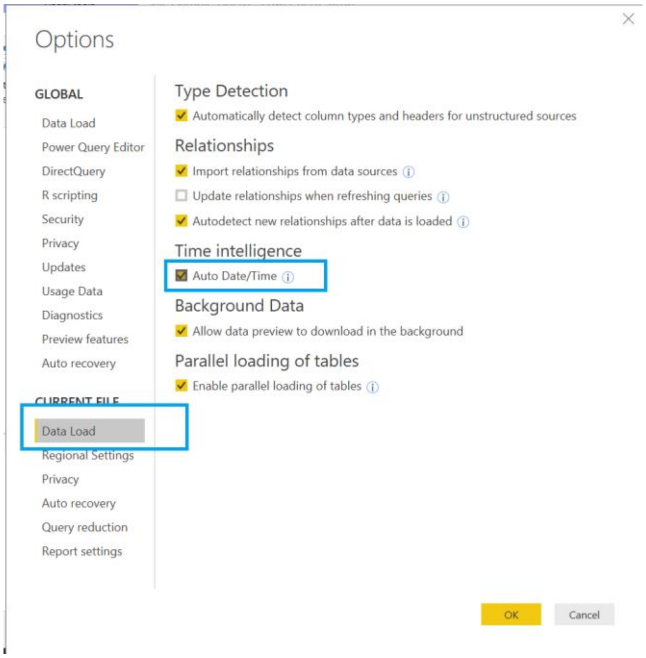
3. Create a new **Table** visual on a new Page and use **TimeNow** column just created. Observe how Year, Quarter, Month and Day gets loaded into different columns automatically!
4. Let us now play around with the time functions now. Go to menu **File** → **Options and settings** → **Options**



Options

Manage options for your Power BI Desktop environment.

Then select the **Data Load** and under **Time intelligence** uncheck the **Auto Date/Time**.



Options

GLOBAL

- Data Load
- Power Query Editor
- DirectQuery
- R scripting
- Security
- Privacy
- Updates
- Usage Data
- Diagnostics
- Preview features
- Auto recovery

CURRENT FILE

- Data Load
- Regional Settings
- Privacy
- Auto recovery
- Query reduction
- Report settings

Type Detection

☑ Automatically detect column types and headers for unstructured sources

Relationships

☑ Import relationships from data sources

☐ Update relationships when refreshing queries

☑ Autodetect new relationships after data is loaded

Time intelligence

☑ Auto Date/Time

Background Data

☑ Allow data preview to download in the background

Parallel loading of tables

☑ Enable parallel loading of tables

Observe the Table visual just created changes and shows only the current date time as one single row. Also observe how the automatically created hierarchy of time got changed to have only oen field now under the fields selection for the Table visual.

5. Change this visual to a **Card** visual.
6. Next create a new measure on bi_date as `MeasureNow = NOW()`
7. Create a new **Card** visual with MeasureNow as the field. Save the report now.
8. **Publish** the report and observe the values of the two visuals. Select on the **Refresh** button and observe what changes in the two visuals!!

Do both the visual values change? If not why? What is the reason behind this change?

Complete this lab by answering the questions above and then checking the **Auto Date/Time** checkbox, save and close the PBIX file.

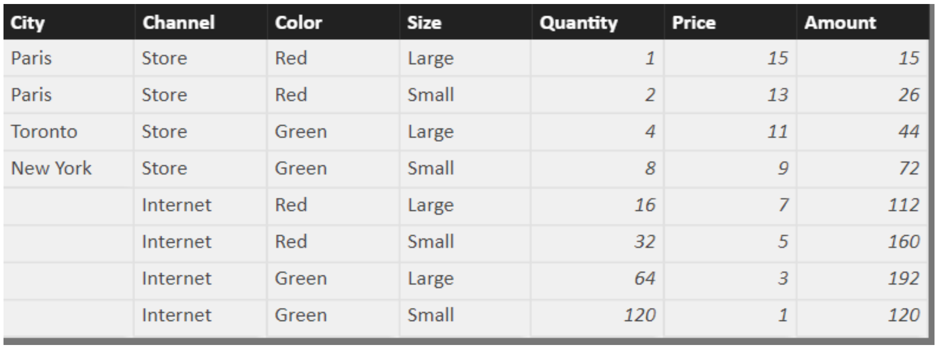
Summary:

With this you have basic understanding of DAX function related to time and how time intelligence makes a difference.

4. Understanding DAX computation logic, Context Transition Lab

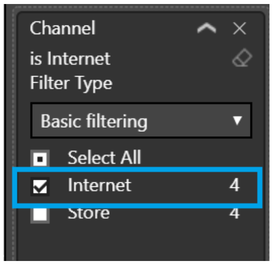
This lab session is aimed at creating and understanding DAX computation in more detail and understanding why things work the way it does. This understanding is if utmost importance with small set of data first and gradually increase the complexity in live projects.

1. Load the Orders.csv file into Power BI desktop. It should appear like the below:



City	Channel	Color	Size	Quantity	Price	Amount
Paris	Store	Red	Large	1	15	15
Paris	Store	Red	Small	2	13	26
Toronto	Store	Green	Large	4	11	44
New York	Store	Green	Small	8	9	72
	Internet	Red	Large	16	7	112
	Internet	Red	Small	32	5	160
	Internet	Green	Large	64	3	192
	Internet	Green	Small	120	1	120

2. Create a page level filter by checking the value of Internet for Channel. This will ensure that only Internet Channel are reported by normal visualization on Power BI report pages.



Channel

is Internet

Filter Type

Basic filtering ▾

■ Select All

☑ Internet 4

□ Store 4

3. Now, create a new measure CalcAmount based on the below DAX:

```
CalcAmount = SUMX(

  FILTER(Orders, Orders[Price] > 1), Orders[Amount])
```

4. Create a new **Table** visual and use the fields – **Color**, **Size**, and **CalcAmount**. Observe that the record line with color as green and size as small does not show any value. Why is that so?
5. Now, create one more measure as

```
CalculateDemo = CALCULATE([CalcAmount], FILTER(ALL(Orders),Orders[Color]="Red"))
```

Include this newly created measure as one of the fields in the above visual. Determine how is the summation happening by looking at the records in the Orders dataset.

6. Now, answer this question: **What is the highest amount for each color in this dataset?**

Think of how this can be achieved. Let us give it a try now.

Create a new measure based on the below DAX expression

```
WithoutContextTransition = MAXX('Orders',

                    SUMX(Orders, Orders[Amount]+0
                    ))
```

7. Now create a **Table** visual to select the **Color** field and the newly created measure **WithoutContextTransition**. Does this give the right answer? Think a while.
8. Next create another measure:

```
WithContextTransition = MAXX('Orders',
                          CALCULATE(SUMX(Orders, Orders[Amount])+0
                        ))
```

9. Create a new **Table** visual and select **Color** and the above newly created measure. Does this give the right answer? **What is the role of CALCULATE in this DAX expression?**
10. Now let us play around with one more DAX function **ALLSELECTED().** Clear the report level filters.
11. Create a new measure:

```
AllSelectedSample = CALCULATE(SUM(Orders[Amount]), ALLSELECTED(Orders[Color]))
```

12. Create a **Table** visual and select as the fields: Color, Channel, and AllSelectedSample

Irrespective of the color, **is the calculation correct for the given Channel?**

13. Next, change the above DAX measure to:

```
AllSelectedSample = CALCULATE(SUM(Orders[Amount]), ALLSELECTED(Orders[Channel]))
```

Is the figure correct for a given color irrespective of the Channel?

14. Observe the values coming because of the changes and the context under which DAX is getting evaluated.

Summary:

With this understanding of context and context transition, you have the understanding of DAX calculations and behavior of certain important functions.