An example using Row Level Security in Power BI with TMVGate

Technical Note: TA2019008

Contents

1.0 Overview

Row Level Security (RLS) in Power BI allows the dataset owner to manage user access. This document illustrates an example of implementing RLS using TMVGate.

In this example, we use the TM1 default sample, 'SalesCube' with a view that returns all the country data in the N level, within the Region dimension. RLS will mainly use Region (Country level data) to perform the mapping to the Power BI users for the access.

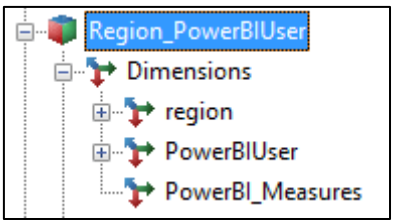
A sample of the dataset in Power BI is shown below:

region	model	actvsbud	account1	month	Value
Argentina	L Series 1.8 L Convertible	Budget	Units	Jan	10
Argentina	L Series 1.8 L Convertible	Budget	Units	Feb	10
Argentina	L Series 1.8 L Convertible	Budget	Units	Mar	10
Argentina	L Series 1.8 L Convertible	Budget	Units	Sep	10
Argentina	L Series 1.8 L Convertible	Budget	Units	Oct	10
Argentina	L Series 1.8 L Convertible	Budget	Units	Nov	10
Belgium	L Series 2.0 L Convertible	Budget	Units	Jan	10
Belgium	L Series 2.0 L Convertible	Budget	Units	Feb	10
Belgium	L Series 2.0 L Convertible	Budget	Units	Mar	10
Belgium	L Series 2.0 L Convertible	Budget	Units	Apr	10
Belgium	L Series 2.0 L Convertible	Budget	Units	Sep	10
Belgium	L Series 2.0 L Convertible	Budget	Units	Oct	10
Belgium	L Series 2.0 L Convertible	Budget	Units	Nov	10
Belgium	L Series 2.0 L Convertible	Budget	Units	Dec	10
Belgium	L Series 2.0 L Wagon	Budget	Units	Feb	10
Belgium	L Series 2.0 L Wagon	Budget	Units	Oct	10
Belgium	L Series 2.0 L Wagon 4WD	Budget	Units	Sep	10
Belgium	L Series 2.0 L Wagon 4WD	Budget	Units	Oct	10
Belgium	L Series 2.5 L Convertible	Budget	Units	Jan	10
Belgium	L Series 2.5 L Convertible	Budget	Units	Feb	10
Belgium	L Series 2.5 L Convertible	Budget	Units	Mar	10
Belgium	L Series 2.5 L Convertible	Budget	Units	Apr	10

2.0 Creating the Mapping Cube between Region and Power BI Users

From a database perspective, there is a need to create a mapping cube as the relationship between Region and Power BI users is a many-to-many relationship. We are assuming that the mapping will not be using the current TM1 user/group information, although technically speaking, this can be done.

A new mapping cube has been defined to handle the mapping of Region to Power BI users. The following shows the cube structure of the Region to PowerBIUser mapping cube.



Dimension Name	Description
Region	All countries (existing dimension)
Power BIUser	Power BI username as elements
Power BI_Measures	A string element to indicate the access

An example of the cube view is shown below. Note that the Access set to “Y” indicates that the specific Power BI User has access to the specific country. Suppress Zero is used when creating the view to be used by TMVGate extraction. This will effectively construct the mapping table in Power BI model.

		PowerBI_Measures
region	PowerBIUser	Access
Denmark		Y
Norway		Y
Sweden		Y
Belgium		Y
		Y
Luxemburg		Y
		Y
Netherlands		Y
		Y
Great Britain		Y
Ireland		Y
France		Y
Germany		Y
Portugal		Y
Spain		Y
Greece		Y
Italy		Y

3.0 Establishing the tables in Power BI

Now with the mapping cube “Region_PowerBIUser”, we are now ready to define the structures in Power BI to set up the RLS.

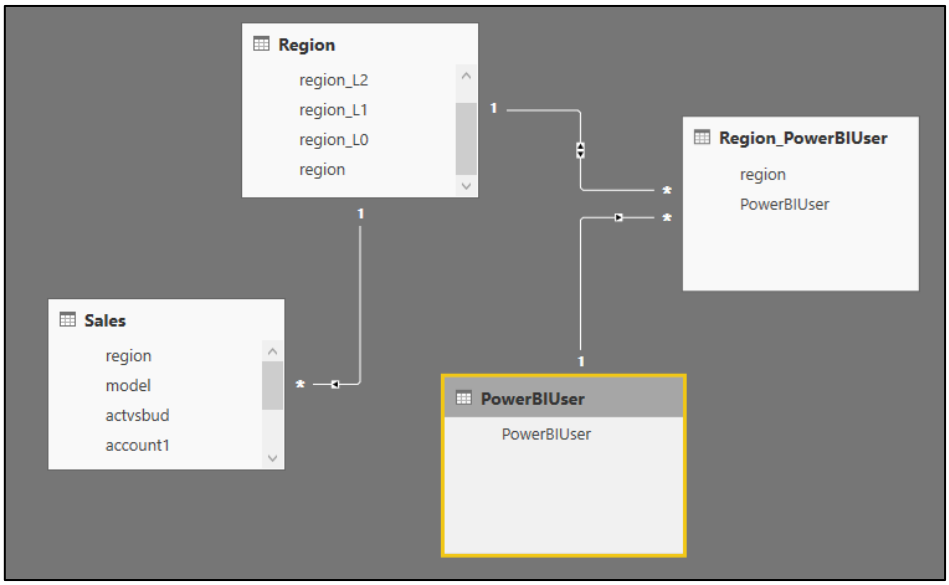
We are assuming that you have already defined the SalesCube date by country as shown in Overview section in Power BI.

Now, generate the TMVGate URL for the “Region_PowerBIUser” cube with a cube view. The data shown in Power BI should be similar to what is shown below. Note that we have deleted the Value column which contains the string “Y”. This is not required for the mapping but rather it’s used together with Suppress Zero to return the access matrix between Region and the Power BI User.

region	PowerBIUser
Denmark	
Norway	
Sweden	
Belgium	
Belgium	
Luxemburg	
Luxemburg	
Netherlands	
Netherlands	
Great Britain	
Ireland	
France	
Germany	
Portugal	
Spain	
Greece	
Italy	

We will need to pull in two more dimensions, namely “Region” and “PowerBIUser” into Power BI. This can be done using the hierarchy extraction. Note that, at the very least, you will need to define a simple hierarchy in “PowerBIUser” in order for the hierarchy extraction to work.

At the end of all TMVGate extractions, you should now have 4 tables in Power BI as shown below:

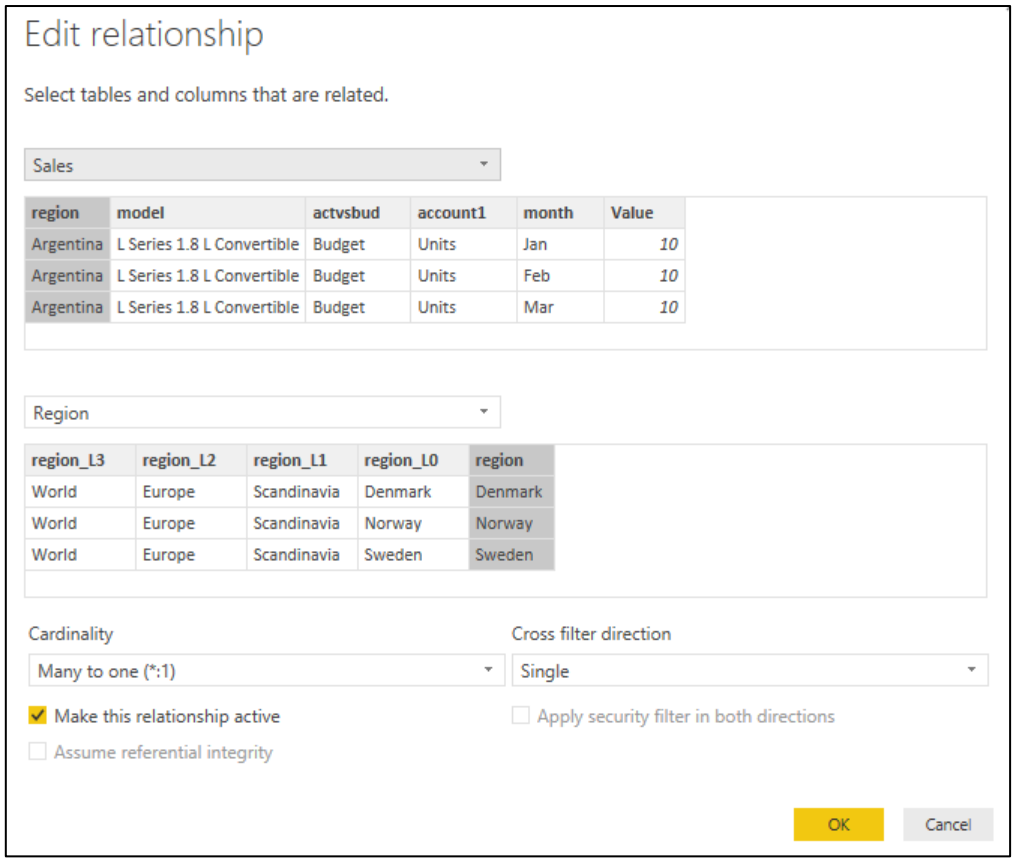


4.0 Linking the tables in Power BI

We will now need to link up the tables. Power BI will auto-detect the relationship for the tables, but we will need to make some adjustment to the relationships.

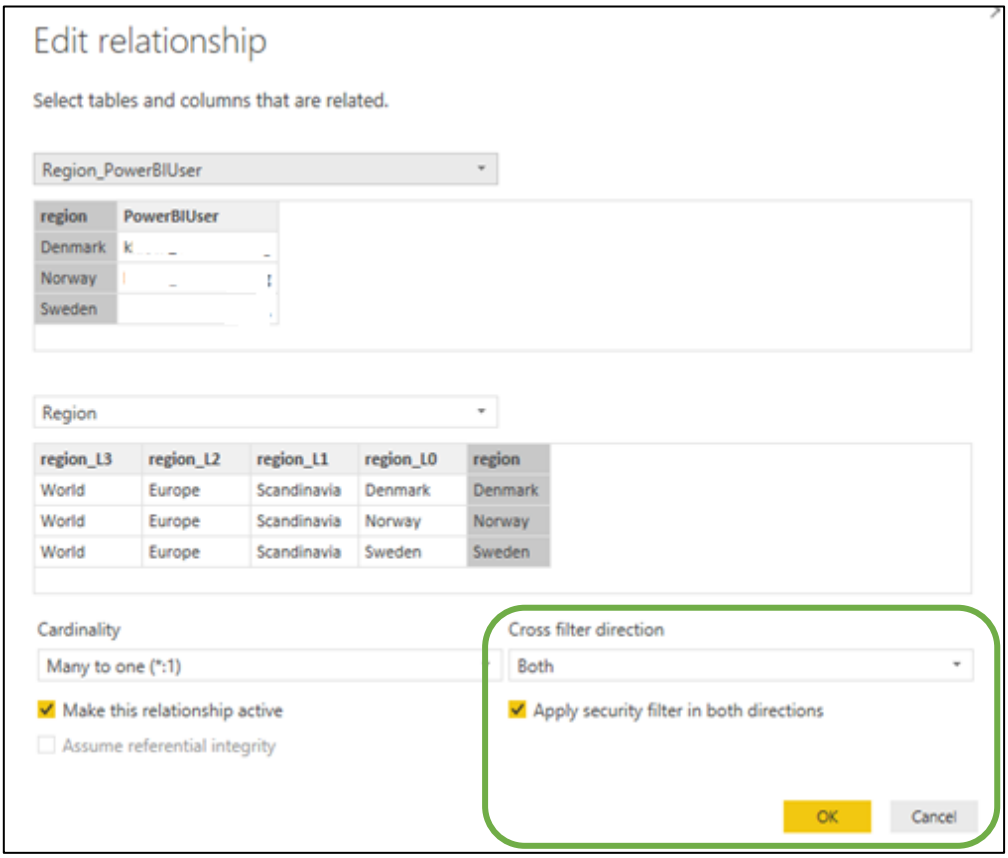
Sales <-> Region

The relationship definition is as shown below:



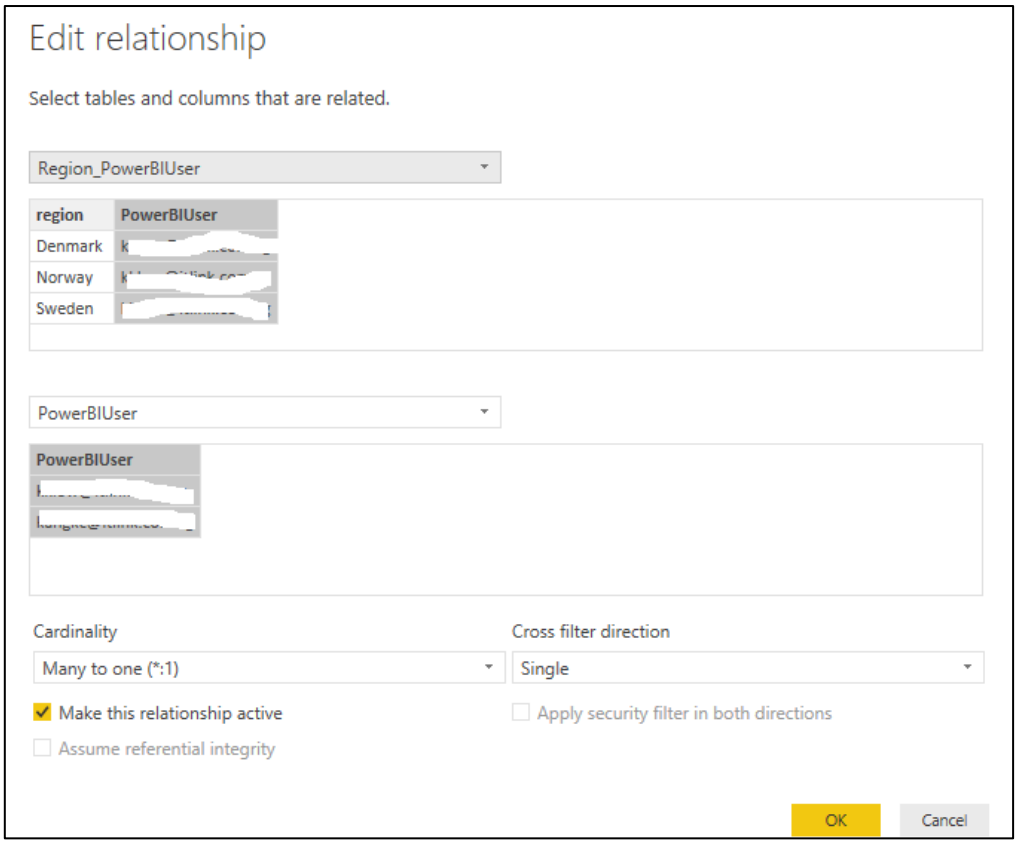
Region_PowerBIUser <-> Region

The relationship definition is as shown below. Note the highlighted in red section. Make sure the security filter is applied in both directions.



Region_PowerBIUser <-> PowerBIUser

The relationship definition is as shown below:

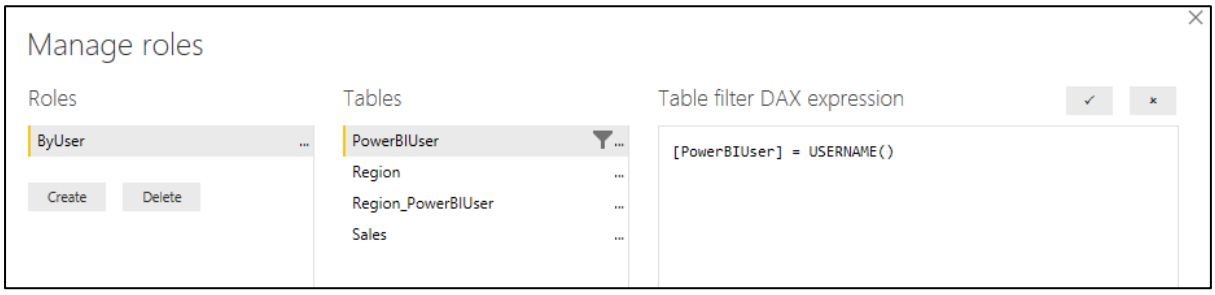


5.0 Define the Security Filter in Power BI

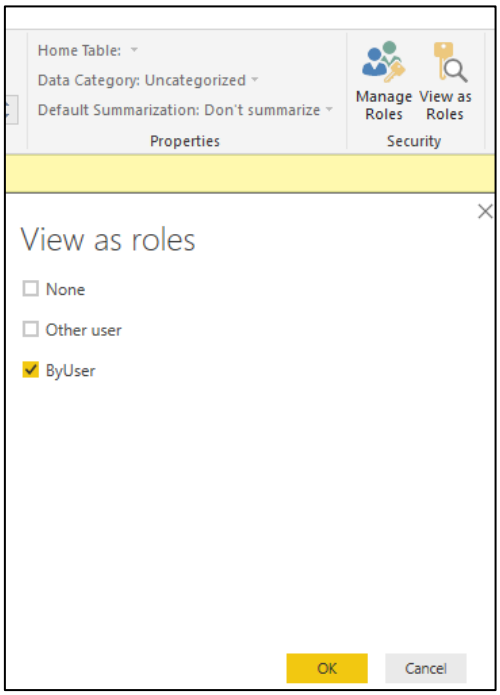
We will now need to define the filter to filter records by Power BI Usernames. Go to Manage Role in Power BI, and define the following filter. Note that we use the DAX function 'USERNAME'. This will return the Power BI login username. In the Power BI Desktop environment, Username returns the SAMAccountname, which is a logon name used to support clients and servers from previous version of Windows (eg domain\username).

This is different in Power BI cloud, where USERNAME() will return the internet-style login name (username@domain). This can be quite confusing, especially during testing.

A quick way to test out the filter is to hardcode the filter to use a specific Power BI Username defined instead of USERNAME(). Once it has been confirmed that the filter is working, you can switch it back before publishing.

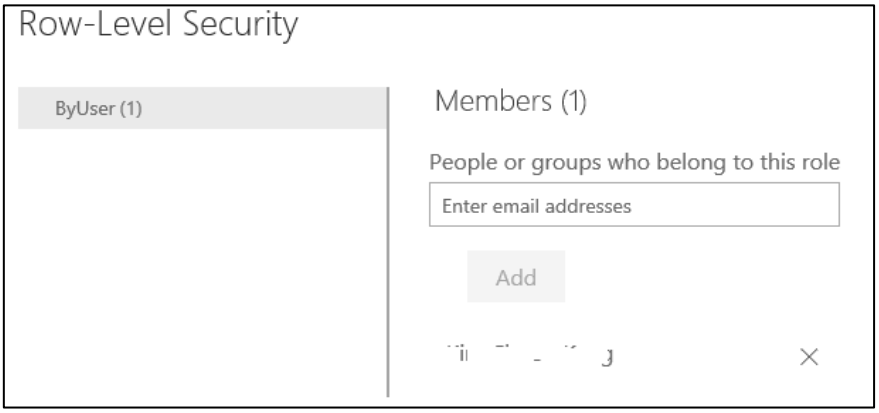


We can test out the filter by using View as Roles as shown below. If the filter is working, it should return only the countries that the specific Power User is defined to have access to.



You can now publish the workbook to Power BI Service for further testing.

Under Power BI Service web interface, you will need to define the members to be included in the published workbook dataset under the role you have defined.



Note that as the owner of the dataset you have published, RLS will not have any effect even if you add yourself as a member.

6.0 Reference Articles on RLS

There are a few good articles on RLS on the Internet which is listed below:

- https://Power BI.microsoft.com/en-us/documentation/Power BI-admin-rls/
- http://radacad.com/dynamic-row-level-security-with-power-bi-made-simple

In addition, the above example is modified from the following blog posts:

- https://www.kasperonbi.com/power-bi-desktop-dynamic-security-cheat-sheet/
- https://community.Power BI.com/t5/Community-Blog/Power-BI-Dynamic-Row-Level-Security-Tips-to-get-it-working/ba-p/76865