

Content

Introduction	2
Enter Data	2 - 3
Implementation	4 – 6

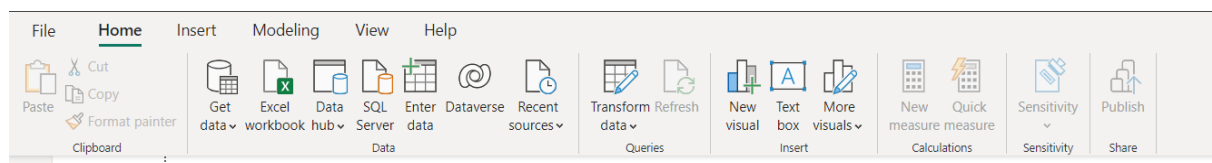
Introduction

There are different methods to use row-level security in Power BI; You can set up Row Level Security in Power BI itself, or through a live connection from a data source such as SSAS Tabular. However, row-level security defined in ways mentioned in the blog posts above isn't dynamic. By dynamic row-level security, I mean the definition of security is beside the user account information in the data source. For example, when John logs in to the system, based on data tables that show John is the sales manager for a specific branch he should be able to see only those branches' data. This method is possible in Power BI using DAX UserName () or UserPrincipalName () function. In this blog post, I'll show you an example of dynamic row-level security with the DAX USERNAME () function in Power BI.

The most important question is why dynamic row-level security? To answer this question, you need to think about the limitation of static row-level security. Static row-level security is simple to implement, however, if you have thousands of roles, then it would be a nightmare to maintain. For example, if you want to create a payroll Power BI report, in a company with ten thousand users, you want every user to have his/her role. Dynamic row-level security is the answer for such scenarios.

Enter Data

1. In this example we are creating two simple tables. Go to the Home page and click on Enter Data option.



2. Now enter the data as shown below and give the table name as Details.

Create Table

	ID	Name	UserName	
1	1	Balaji	lacchi.balaji@gisul.co.in	
2	2	Hari	Hari.v@gisul.co.in	
3	3	Ram	Ram.P@gisul.co.in	
+				

Name:

Load

Edit

Cancel

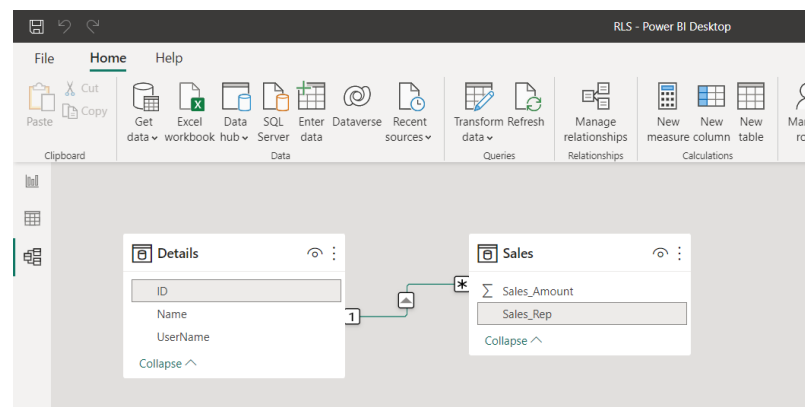
- Click on Load.
- Create another table with the same process as shown below.

	Sales_Amount	Sales_Rep	
1	2000	1	
2	3000	2	
3	5000	3	
4	7000	1	
+			

Name:

Implementation

- Go to model area and create a relationship between two tables based on Id and Sales_Rep.



- Create a table with the following fields as shown below.

Name	UserName	Sales_Amount
Balaji	lacchi.balaji@gisul.co.in	9000
Hari	Hari.v@gisul.co.in	3000
Ram	Ram.P@gisul.co.in	5000
Total		17000

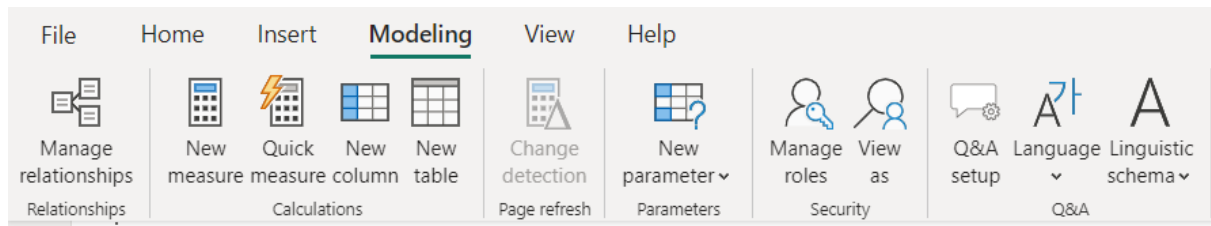
Columns

- Name
- UserName
- Sales_Amount

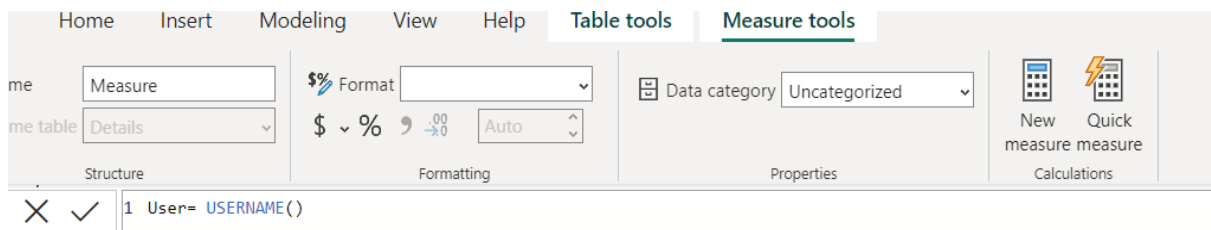
Drill through

Cross-report ☐

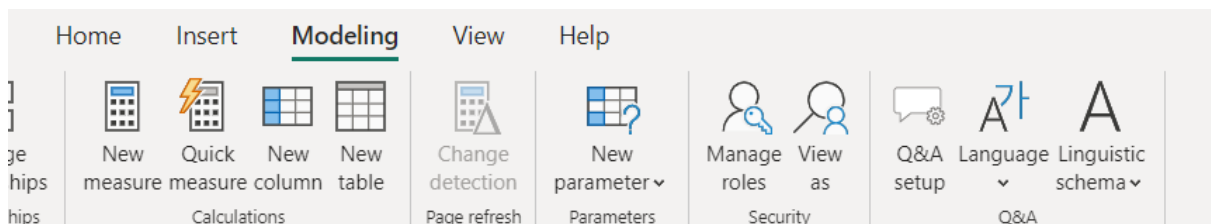
- Now we need to create a measure. To create it go to modelling and click on New Measure as shown below.



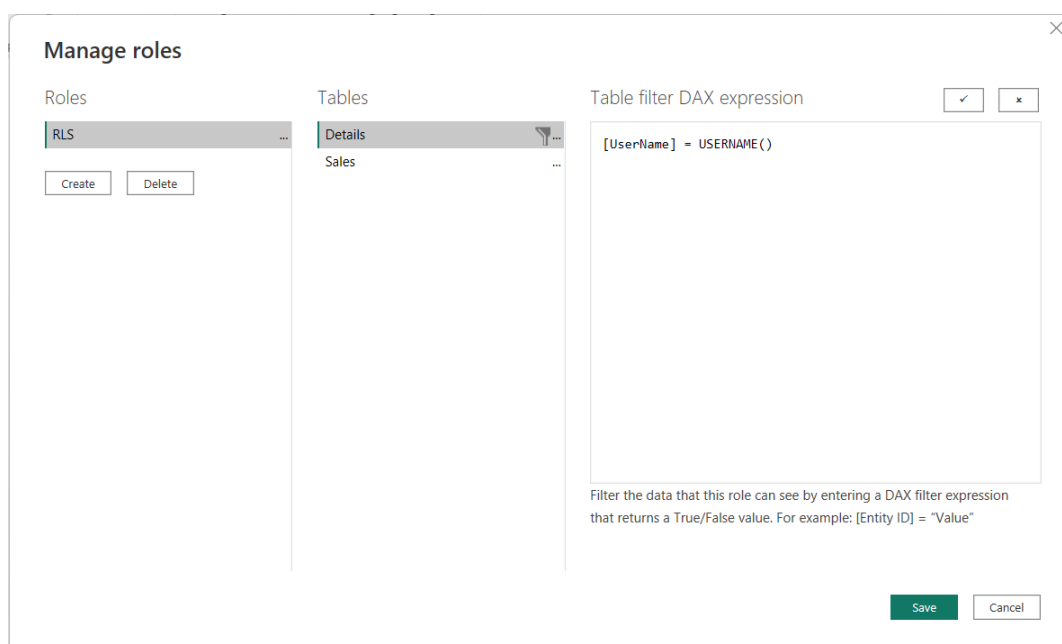
- Add a measure for USERNAME () in DAX to see the user logged in from my report.
- So, in Data Tab, create a new measure, and name it User, with a value of USERNAME ();



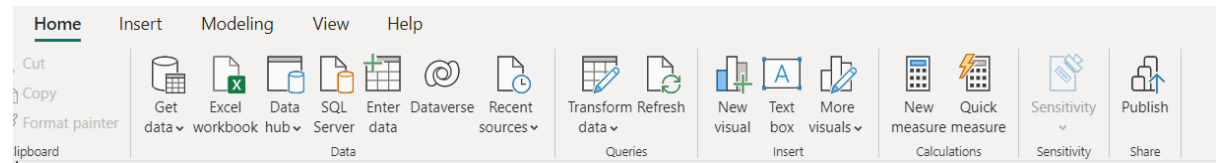
- Now we need to create a Role. So go to modeling and click on Manage role as shown below.



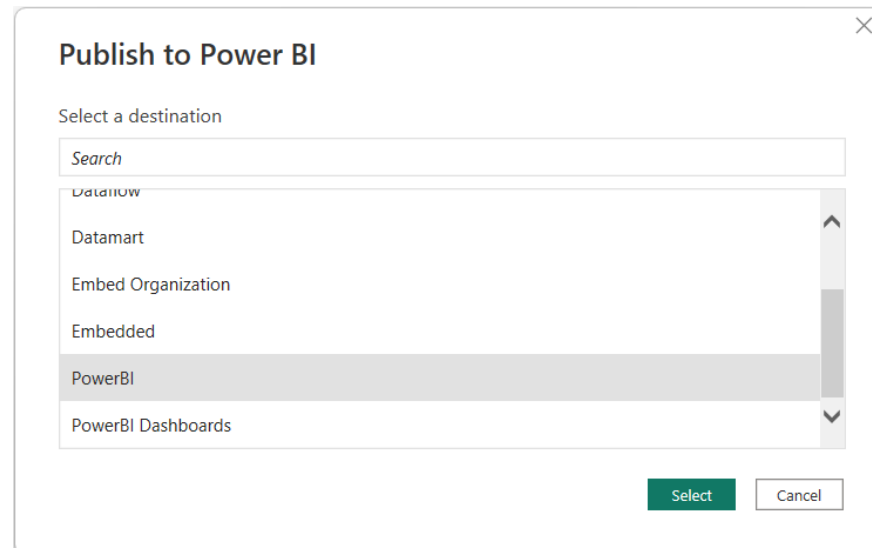
- Here I will only use that technique to filter each role based on their username with the DAX username () function.
- To create security, go to the Modeling tab, and Manage Roles. Create a role and name it RLS. and define a filter on the Sales table as below.



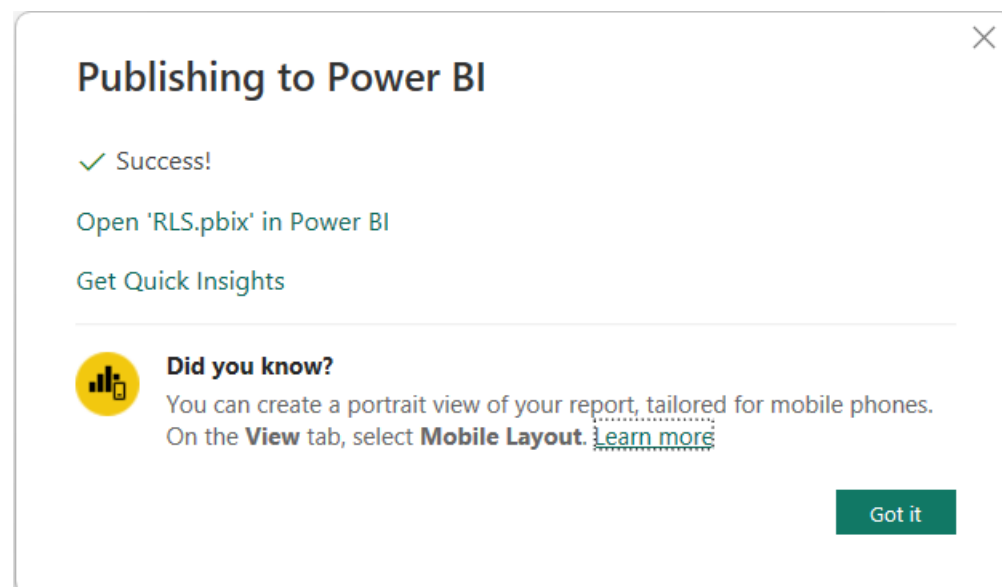
13. Now save our power bi file into our system.
14. Now we need to publish the file into the power bi service.
15. Click on the Publish button as shown below.



16. Select the workspace and click on select.

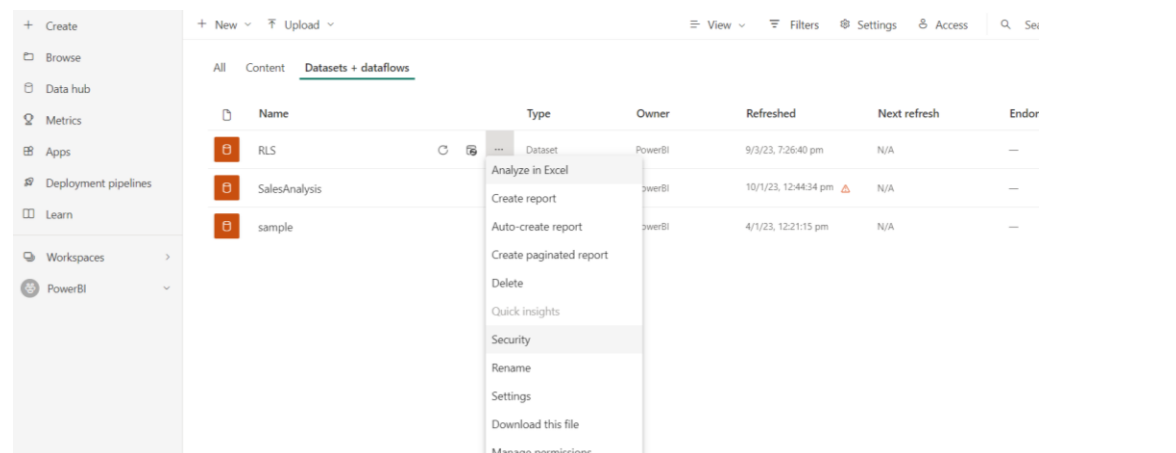


17. We will get a successful notification in the tab as shown below.

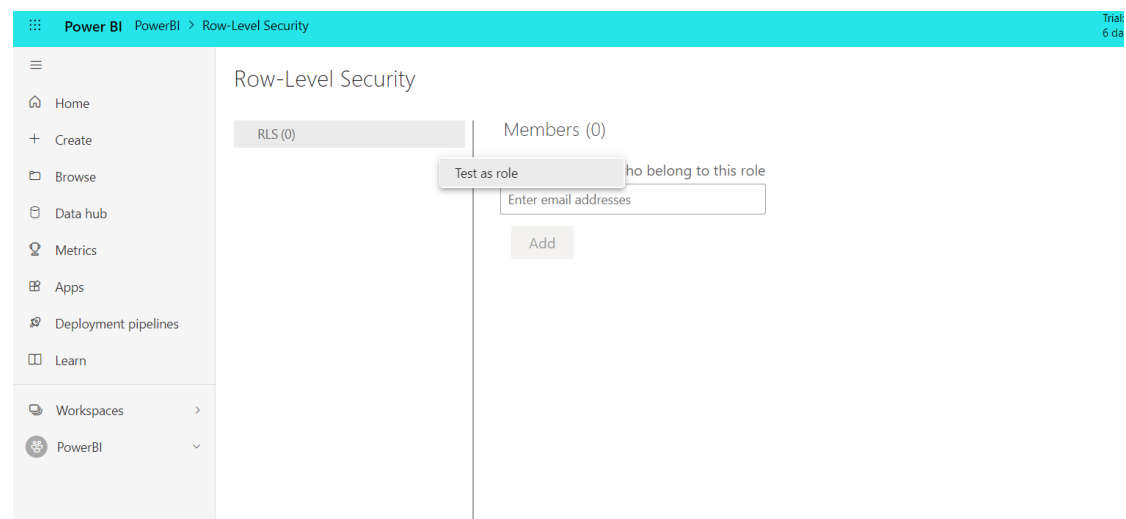


18. Now go to the power bi service and open the workspace where we publish our file.

19. Then go to the Datasets+Dataflow option. Right-click and click on security as shown below.



20. Now you will see an RLS that we have created right click on it and click on Test as role as shown below.



21. if the usernames match one of the entries in the Details table, they would see their names, and data rows related to that in the report.

22. As you can see it is only my record in the report.

