
Debabrata Palit
/debabrata-palit03

Row-Level Security

TRANSFORM YOUR DATA INTO DECISIONS!

Introduction

Row-level security (RLS) allows you to restrict data access to specific users or groups based on defined rules, effectively filtering data visibility at the row level within tables.

Filters restrict data access at the row level, and you can define filters within roles. In the Power BI service, users with access to a workspace have access to semantic models in that workspace.

RLS only restricts data access for users with Viewer permissions. It doesn't apply to Admins, Members, or Contributors.

Key Aspects

Restricting data access:
RLS helps ensure that users only see the data relevant to them, enhancing data security and privacy.

Defining roles and rules:
You can define roles within Power BI Desktop and assign users or groups to these roles.

Using DAX:
DAX expressions are used to create filters that determine which rows a user can see based on their role.

Dynamic and Static RLS:
You can implement both static RLS, where rules are fixed, and dynamic RLS, where rules are based on user-specific information like their email or username.

Integration with Power BI service:
Once defined in Power BI Desktop, roles and rules are published to the Power BI service, where you can assign users to the roles.

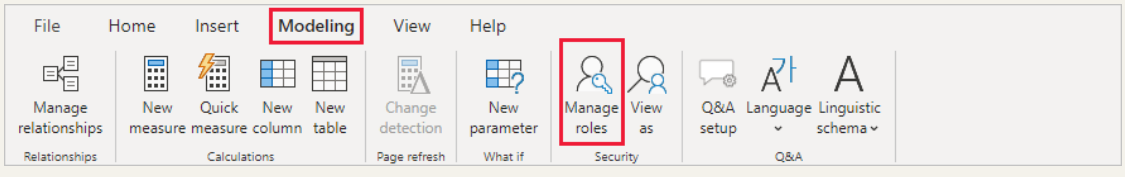

Debabrata Palit
/debabrata-palit03

Define Roles & Rules

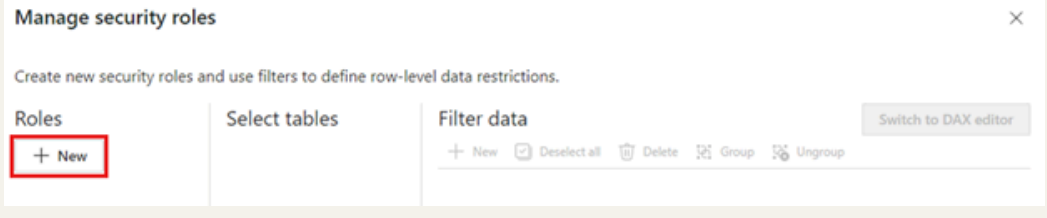
1. Import data into your Power BI Desktop report, or configure a DirectQuery connection.

[Note] You can't define roles within Power BI Desktop for Analysis Services live connections. You need to do that within the Analysis Services model.

2. From the **Modeling** tab, select **Manage Roles**.

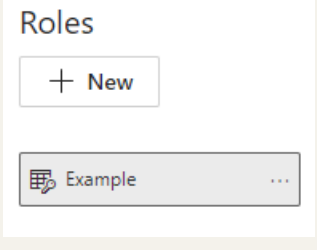


3. From the Manage roles window, select **New** to create a new role.



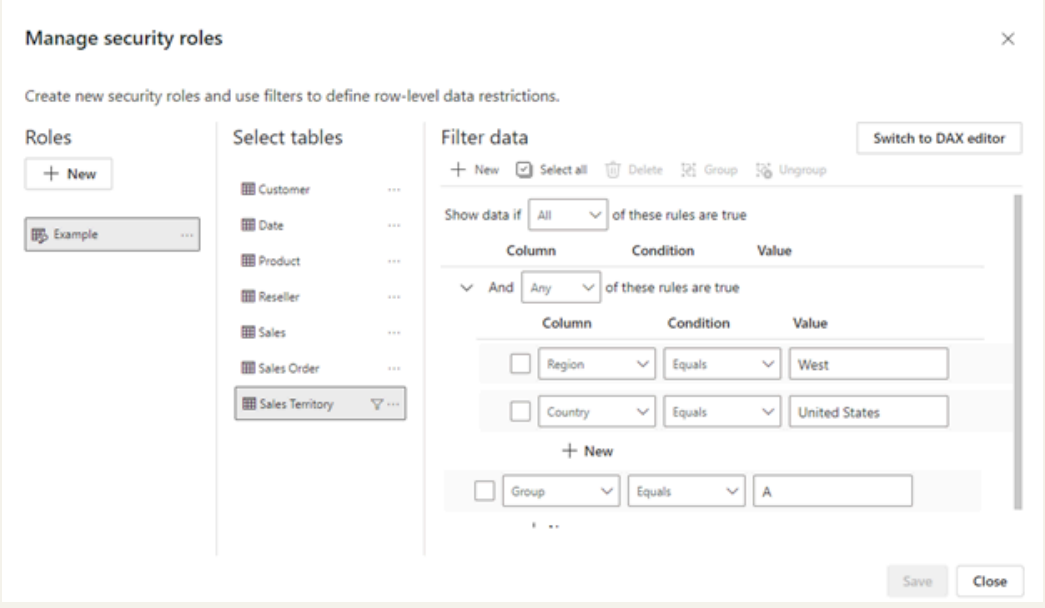
4. Under Roles, provide a name for the role and select enter.

[Note]: You can't define a role with a comma, for example *London,ParisRole*.

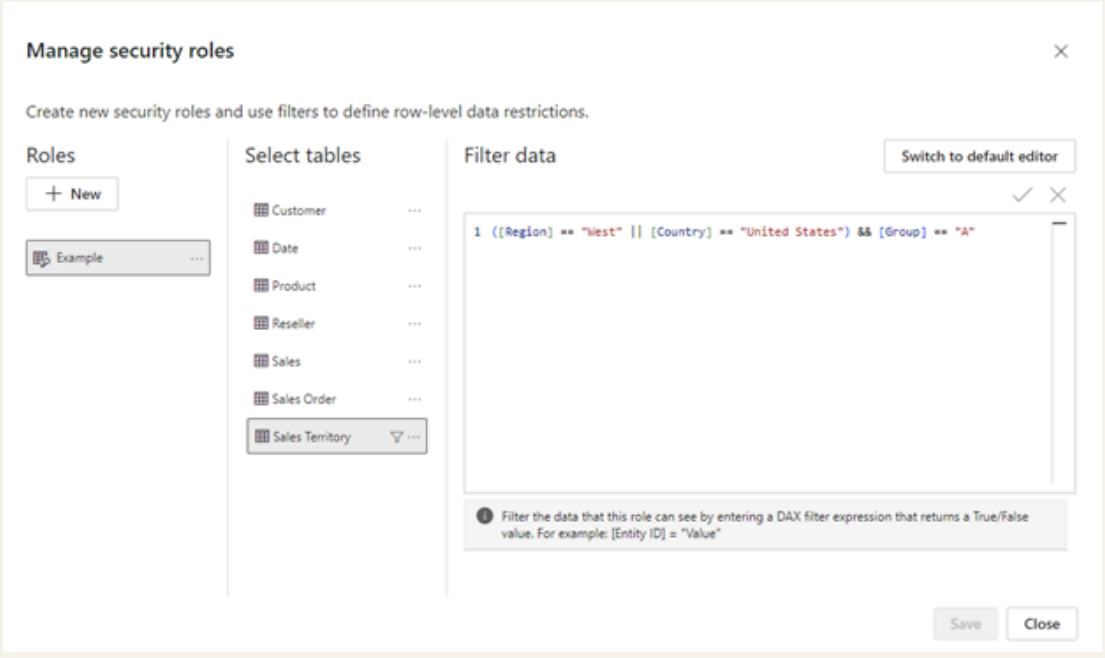


5. Under Select tables, select the table to which you want to apply a row-level security filter.

6. Under Filter data, use the default editor to define your roles. The expressions created return a true or false value.



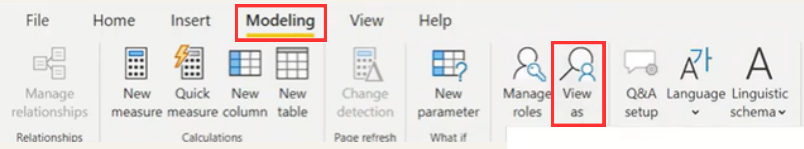
7. Optionally select Switch to DAX editor to switch to using the DAX editor to define your role. DAX expressions return a value of true or false. For example: *[Entity ID] = "Value"*. The DAX editor is complete with autocomplete for formulas (intellisense). You can select the checkmark above the expression box to validate the expression and the X button above the expression box to revert changes.



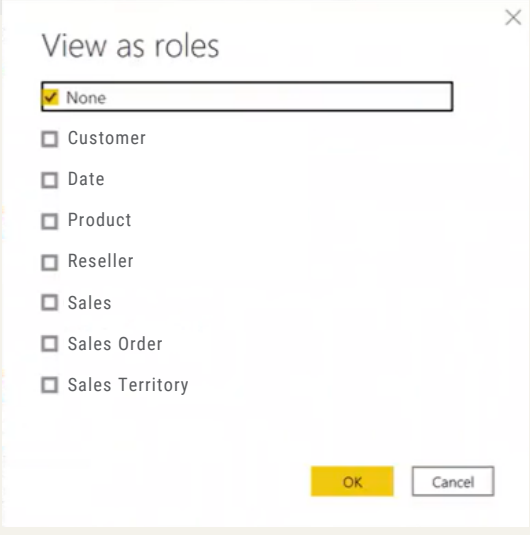
8. Select **Save**.

[Note]: You can't assign users to a role within Power BI Desktop. You assign them in the Power BI service. You can enable dynamic security within Power BI Desktop by making use of the *username()* or *userprincipalname()* DAX functions and having the proper relationships configured.

Within Power BI Desktop, *username()* will return a user in the format of *DOMAIN\User* and *userprincipalname()* will return a user in the format of *user@contoso.com*. Within the Power BI service, *username()* and *userprincipalname()* will both return the user's User Principal Name (UPN).



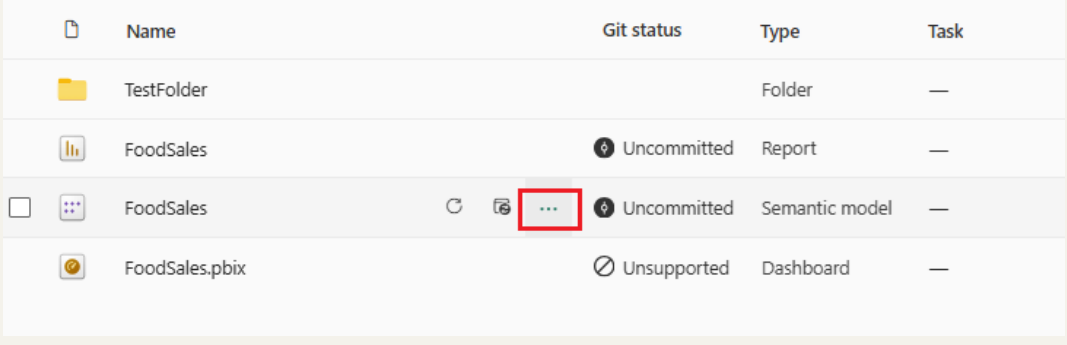
[Validate]:
Select **Modeling** –> **View As** –> Select the role you just set up –> Select **Ok** –> Then verify that the data is filtering as expected.

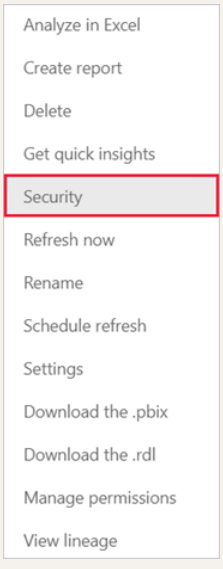


9. **Save** and **publish** the Power BI Report to a workspace.

Now, Manage the Security on your Model:

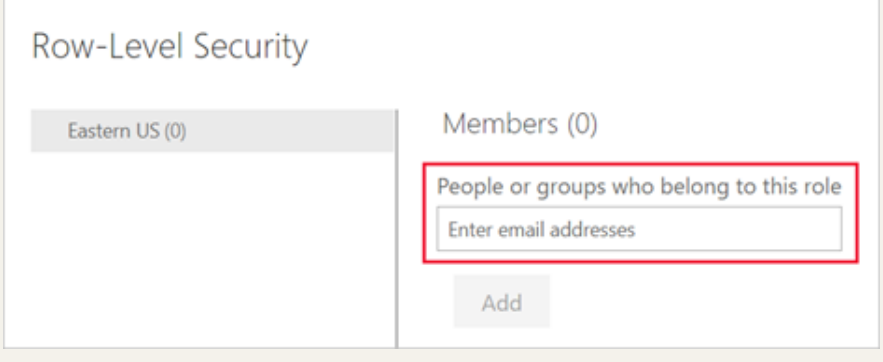
10. Now go to Fabric, select the **More options (...)** menu for a semantic model –> Select **Security**.

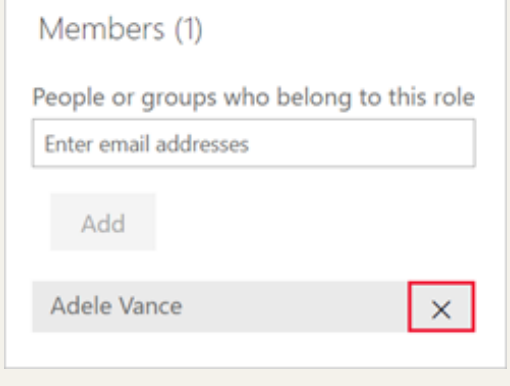




Security takes you to the Role-Level Security page where you add members to a role you created. Contributor (and higher workspace roles) will see Security and can assign users to a role.

[Add Members]:
In the Power BI service, you can add a member to the role by typing in the email address or name of the user or security group. You can't add Groups created in Power BI. You can add members external to your organization.



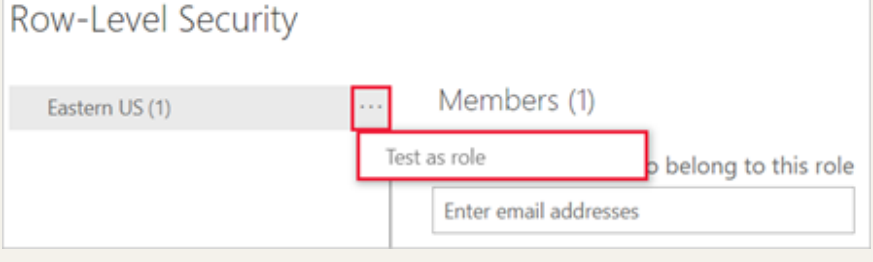


[Remove Members]:
You can remove members by selecting the ✕ next to their name.

[Validating the role]:
You can validate that the role you defined is working correctly in the Power BI service by testing the role.

1. Select **More options (...)** next to the role.
2. Select **Test as role**.
3. Ensure that the RLS is working as expected.

You're redirected to the report that was published from Power BI Desktop with this semantic model, if it exists. Dashboards aren't available for testing using the **Test as role** option.



Types of RLS

Static RLS:

Static RLS defines fixed access rules for specific roles. These rules don't change based on the user's identity.

How it works: You create roles in Power BI Desktop and define DAX filters for each role, specifying which data rows that role can access. For example, a role "North Region" might have a filter [Region] = "North".

Use cases: Static RLS is suitable when you have a limited number of users with clearly defined roles and stable access requirements that don't change frequently.

Advantages:

- Easy to set up and implement.
- Works well for small datasets with fixed security rules.
- Simplified management for a static set of access rules.

Limitations:

- Not very flexible for complex access rules or frequent changes.
- Requires creating separate roles for each user group, which can be cumbersome with many roles.
- Requires manual updates when user access changes.

Dynamic RLS:

Dynamic RLS filters data based on the logged-in user, using their user principal name (UPN) or other attributes to determine data access.

How it works: You typically create a security table that maps users (e.g., UPNs) to the data they can access (e.g., specific regions or departments). You then create roles with DAX expressions that use functions like USERPRINCIPALNAME() to filter data based on the current user's UPN.

Use cases: Dynamic RLS is more appropriate for scenarios with a large number of users, frequently changing access requirements, or complex rules that need to adapt automatically.

Advantages:

- Scalable for large numbers of users and different roles.
- Flexible and adaptable to complex access rules and changing user attributes.
- Can automate access updates when user roles or departments change.

Limitations:

- More complex to set up than static RLS.
- May require more effort to troubleshoot if there are issues with access permissions.

THANK YOU!!!

STAY CONNECTED

SHARE YOUR THOUGHTS

SAVE FOR LATER

LIKE THIS POST