

Row-level security (RLS) with Power BI

Row-level security (RLS) with Power BI can be used to restrict data access for given users. Filters restrict data access at the row level, and you can define filters within roles. In the Power BI service, members of a workspace have access to datasets in the workspace. RLS doesn't restrict this data access.

You can configure RLS for data models imported into Power BI with Power BI Desktop. You can also configure RLS on datasets that are using DirectQuery, such as SQL Server. For Analysis Services or Azure Analysis Services live connections, you configure Row-level security in the model, not in Power BI Desktop. The security option will not show up for live connection datasets.

Define roles and rules in Power BI Desktop

You can define roles and rules within Power BI Desktop. When you publish to Power BI, it also publishes the role definitions.

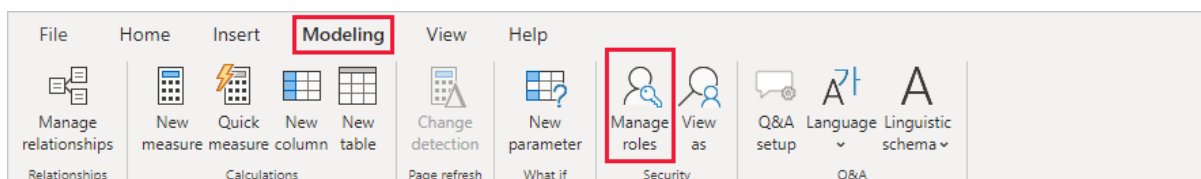
To define security roles, follow these steps.

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

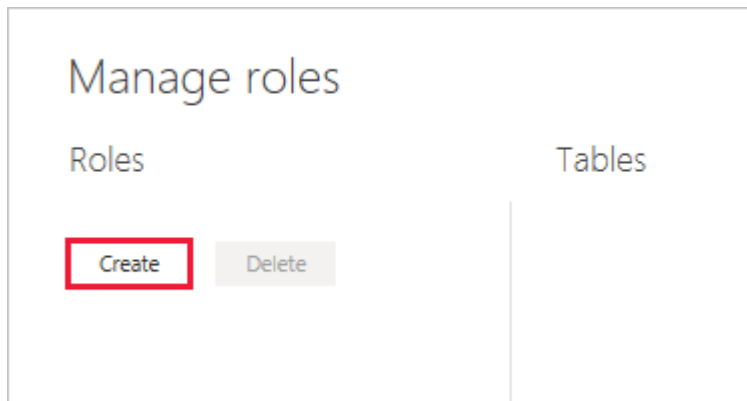
Note

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 **Create**.

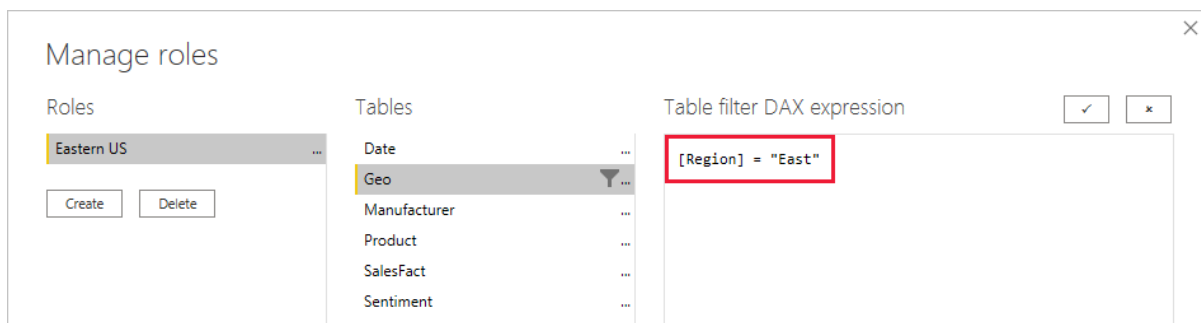


4. Under **Roles**, provide a name for the role.

Note

You can't define a role with a comma, for example London,ParisRole.

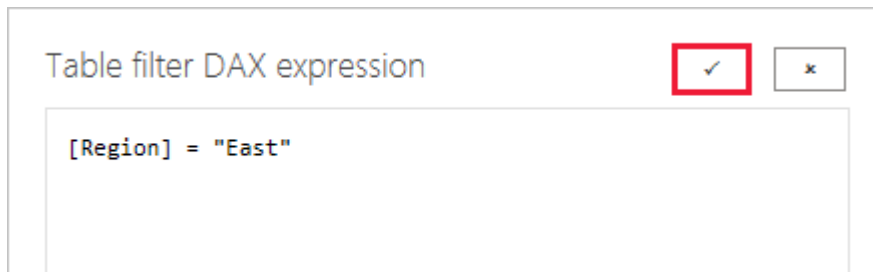
5. Under **Tables**, select the table to which you want to apply a DAX rule.
6. In the **Table filter DAX expression** box, enter the DAX expressions. This expression returns a value of true or false. For example: [Entity ID] = "Value".



Note

You can use `username()` within this expression. Be aware that `username()` has the format of `DOMAIN\username` within Power BI Desktop. Within the Power BI service and Power BI Report Server, it's in the format of the user's User Principal Name (UPN). Alternatively, you can use `userprincipalname()`, which always returns the user in the format of their user principal name, `username@contoso.com`.

7. After you've created the DAX expression, select the checkmark above the expression box to validate the expression.



Note

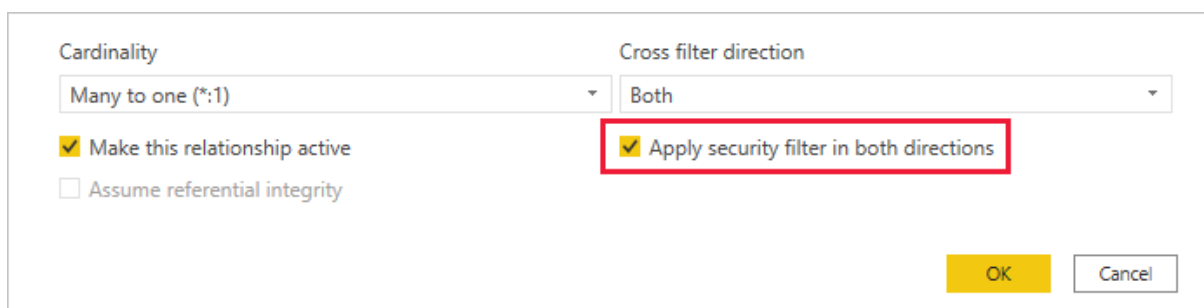
In this expression box, you use commas to separate DAX function arguments even if you're using a locale that normally uses semicolon separators (e.g. French or German).

8. Select **Save**.

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.

By default, row-level security filtering uses single-directional filters, whether the relationships are set to single direction or bi-directional. You can manually enable bi-directional cross-filtering with row-level security by selecting the relationship and checking the **Apply security filter in both directions** checkbox. Note that if a table takes part in multiple bi-directional relationships you can only select this option for one of those relationships. Select this option when you've also implemented dynamic row-level security at the server level, where row-level security is based on username or login ID.

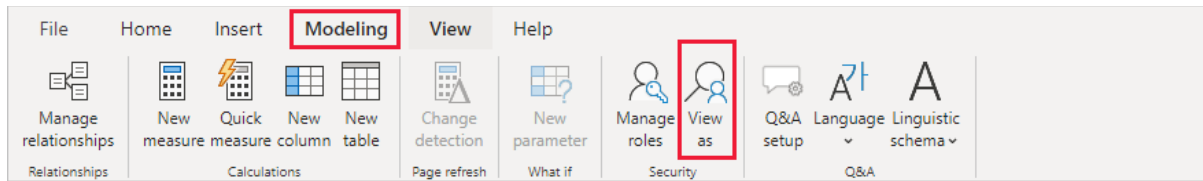
For more information, see [Bidirectional cross-filtering using DirectQuery in Power BI Desktop](#) and the [Securing the Tabular BI Semantic Model](#) technical article.



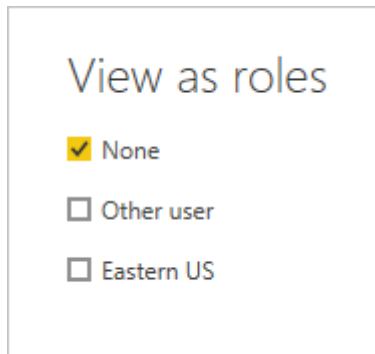
Validate the roles within Power BI Desktop

After you've created your roles, test the results of the roles within Power BI Desktop.

1. From the **Modeling** tab, select **View as**.



The **View as roles** window appears, where you see the roles you've created.



2. Select a role you created, and then select **OK** to apply that role.

The report renders the data relevant for that role.

3. You can also select **Other user** and supply a given user.



It's best to supply the User Principal Name (UPN) as that's what the Power BI service and Power BI Report Server use.

Within Power BI Desktop, **Other user** displays different results only if you're using dynamic security based on your DAX expressions.

4. Select **OK**.

The report renders based on what that user can see.

Note

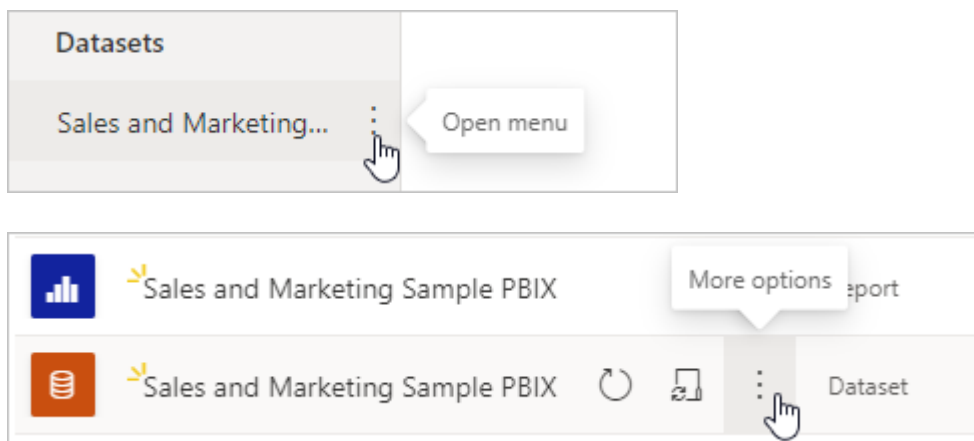
The View as role feature doesn't work for DirectQuery models with Single Sign-On (SSO) enabled.

Now that you're done validating the roles in Power BI Desktop, go ahead and publish your report to the Power BI service.

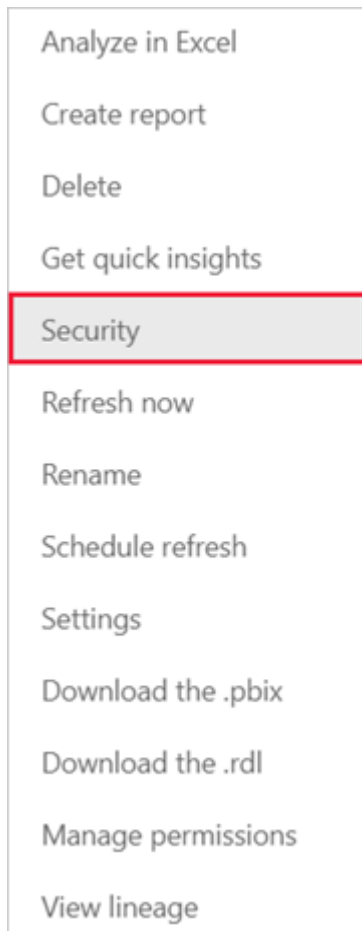
Manage security on your model

To manage security on your data model, open the workspace where you saved your report in the Power BI service and do the following steps:

1. In the Power BI service, select the **More options** menu for a dataset. This menu appears when you hover on a dataset name, whether you select it from the navigation menu or the workspace page.



2. Select **Security**.



Security will take you to the Role-Level Security page where you add members to a role you created in Power BI Desktop. Contributor (and higher workspace roles) will see **Security** and can assign users to a role.

You can only create or modify roles within Power BI Desktop.

Working with members

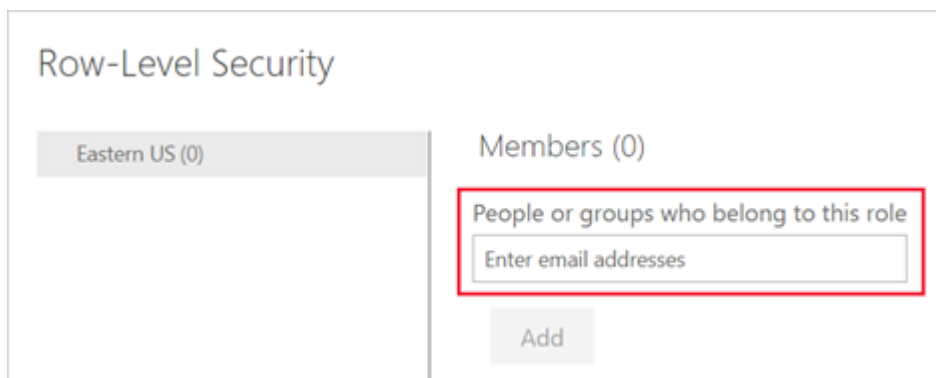
Add members

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](#).

You can use the following groups to set up row level security.

- Distribution Group
- Mail-enabled Group
- Security Group

Note, however, that Office 365 groups are not supported and cannot be added to any roles.



Row-Level Security

Eastern US (0)

Members (0)

People or groups who belong to this role

Enter email addresses

Add

You can also see how many members are part of the role by the number in parentheses next to the role name, or next to Members.



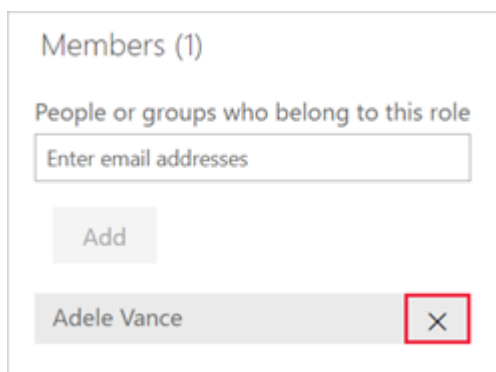
Row-Level Security

Eastern US (1)

Members (1)

Remove members

You can remove members by selecting the X next to their name.



Members (1)

People or groups who belong to this role

Enter email addresses

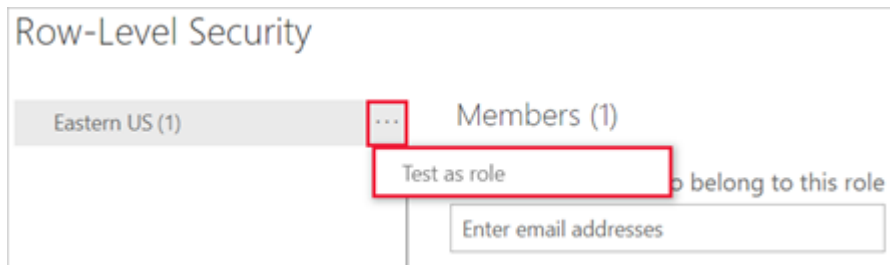
Add

Adele Vance X

Validating the role within the Power BI service

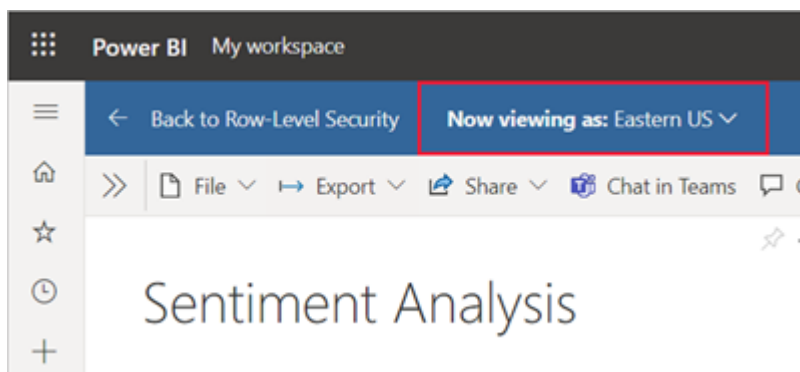
You can validate that the role you defined is working correctly in the Power BI service by testing the role.

1. Select **More options** (...) next to the role.
2. Select **Test data as role**.

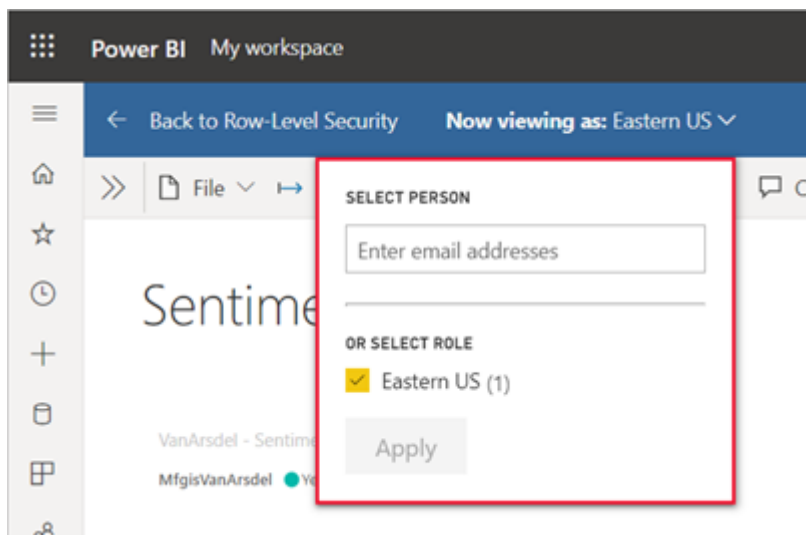


You'll be redirected to the report that was published from Power BI Desktop with this dataset, if it exists. Dashboards are not available for testing using the **Test as role** option.

In the page header, the role being applied is shown.



Test other roles, or a combination of roles, by selecting **Now viewing as**.



You can choose to view data as a specific person or you can select a combination of available roles to validate they're working.

To return to normal viewing, select **Back to Row-Level Security**.

Note

The Test as role feature doesn't work for DirectQuery models with Single Sign-On (SSO) enabled.

Using the `username()` or `userprincipalname()` DAX function

You can take advantage of the DAX functions `username()` or `userprincipalname()` within your dataset. You can use them within expressions in Power BI Desktop. When you publish your model, it will be used within the Power BI service.

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). This looks similar to an email address.

Using RLS with workspaces in Power BI

If you publish your Power BI Desktop report to a [workspace](#) in the Power BI service, the RLS roles are applied to members who are assigned to the **Viewer** role in the workspace. Even if **Viewers** are given Build permissions to the dataset, RLS still applies. For example, if Viewers with Build permissions use [Analyze in Excel](#), their view of the data will be protected by RLS. Workspace members assigned **Admin**, **Member**, or **Contributor** have edit permission for the dataset and, therefore, RLS doesn't apply to them. If you want RLS to apply to people in a workspace, you can only assign them the **Viewer** role. Read more about [roles in workspaces](#).

Considerations and limitations

The current limitations for row-level security on cloud models are as follows:

- If you previously defined roles and rules in the Power BI service, you must re-create them in Power BI Desktop.
- You can define RLS only on the datasets created with Power BI Desktop. If you want to enable RLS for datasets created with Excel, you must convert your files into Power BI Desktop (PBIX) files first. [Learn more](#).
- Service principals cannot be added to an RLS role. Accordingly, RLS won't be applied for apps using a service principal as the final effective identity.

- Only Import and DirectQuery connections are supported. Live connections to Analysis Services are handled in the on-premises model.
- The Test as role/View as role feature doesn't work for DirectQuery models with Single Sign-On (SSO) enabled.

FAQ

Question: What if I had previously created roles and rules for a dataset in the Power BI service? Will they still work if I do nothing?

Answer: No, visuals will not render properly. You will have to re-create the roles and rules within Power BI Desktop and then publish to the Power BI service.

Question: Can I create these roles for Analysis Services data sources?

Answer: You can if you imported the data into Power BI Desktop. If you are using a live connection, you will not be able to configure RLS within the Power BI service. This is defined within the Analysis Services model on-premises.

Question: Can I use RLS to limit the columns or measures accessible by my users?

Answer: No, if a user has access to a particular row of data, they can see all the columns of data for that row.

Question: Does RLS let me hide detailed data but give access to data summarized in visuals?

Answer: No, you secure individual rows of data but users can always see either the details or the summarized data.

Question: My data source already has security roles defined (for example SQL Server roles or SAP BW roles). What is the relationship between these and RLS?

Answer: The answer depends on whether you're importing data or using DirectQuery. If you're importing data into your Power BI dataset, the security roles in your data source aren't used. In this case, you should define RLS to enforce security rules for users who connect in Power BI. If you're using DirectQuery, the security roles in your data source are used. When a user opens a report Power BI sends a query to the underlying data source, which applies security rules to the data based on the user's credentials.