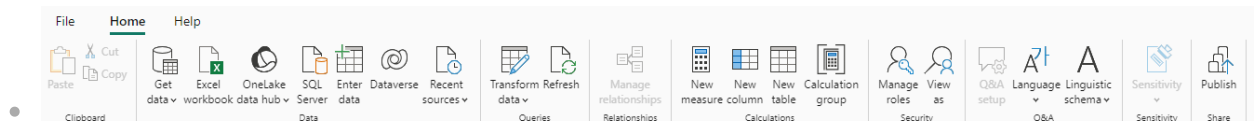


Explain the use of Static RLS and Dynamic RLS

Implement Static RLS

1. Create Roles in Power BI Desktop

- Open the Power BI Desktop report.
- Go to the **Modeling** tab and click **Manage Roles** Button on the upper Bar.



- Create a new role and define a **DAX filter** on tables to restrict data.
 - Example: `Sales[Region] = "West"` (this will restrict data to only the "West" region).

2. Assign Users to Roles

- Publish the report to the Power BI Service.
- <https://app.powerbi.com>
- In Power BI Service, go to the dataset settings and assign users to specific roles.

3. Testing Static RLS

- After defining roles and publishing the report, you can test it by going to **Modeling > View As** and selecting the role to ensure it works as expected.

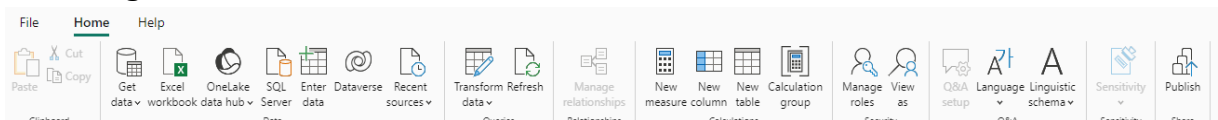
Implement Dynamic RLS

1. Create a Security Table

- In your data model, create a **security table** that maps users to specific data filters (e.g., a table that links users' email addresses to their region or department).
 - Example: A table called `UserRoles` with columns like `UserEmail` and `Region`.
 - Example: 3 tables, 2 of them showing data and having column `Country` and third that a security table that contains the intersection of those two and set as security

2. Define DAX Filter Using the Security Table

- In **Manage Roles**,



create a DAX filter that uses the security table to filter data based on the logged-in user.

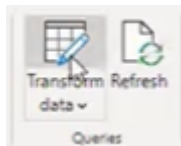
- Example: `Sales[Region] = LOOKUPVALUE(UserRoles[Region], UserRoles[UserEmail], USERPRINCIPALNAME())`
- The `USERPRINCIPALNAME()` function returns the logged-in user's email, which you can use to match the security table.

3. Test Dynamic RLS

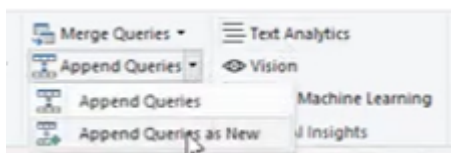
- Use **View As Roles** in Power BI Desktop to test how different users will see the data. Enter the email address of a user to check if the dynamic filter is applied correctly.

Name	Last Name	Title	Country	Email
Jenny	McFarland	Sr Analyst	France	Jenny.McFarland@company.com
Mark	Lopez	Vice President	France	Mark.Lopez@company.com
Mark	Lopez	Vice President	England	Mark.Lopez@company.com
Mark	Lopez	Vice President	Germany	Mark.Lopez@company.com
Mark	Lopez	Vice President	Italy	Mark.Lopez@company.com
Mark	Lopez	Vice President	Spain	Mark.Lopez@company.com
Martha	Gosney	Sr Analyst	England	Martha.Gosney@company.com
Nestor	Adrianzen	Manager	Germany	nestor@vizxlization.onmicrosoft.com
Jose	Davis	Manager	Italy	Jose.Davis@company.com
John	Miller	Sr Analyst	Spain	John.Miller@company.com

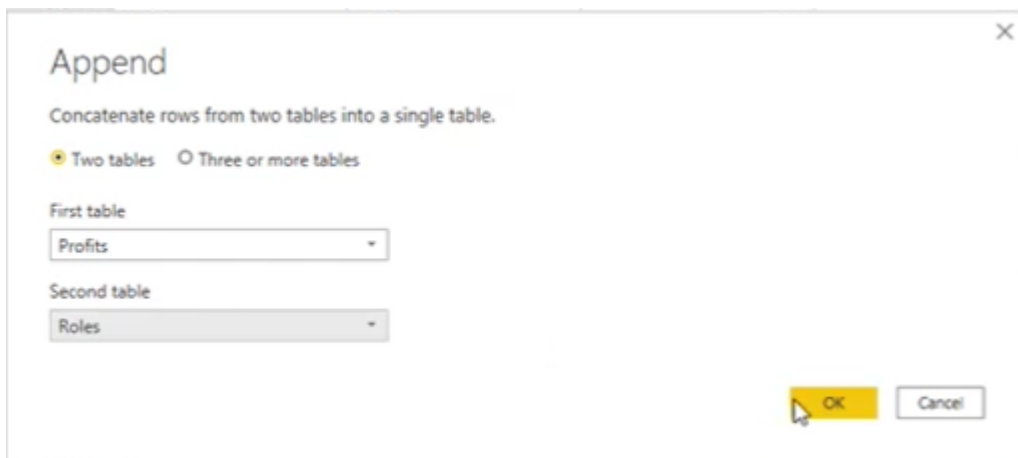
4.



5.

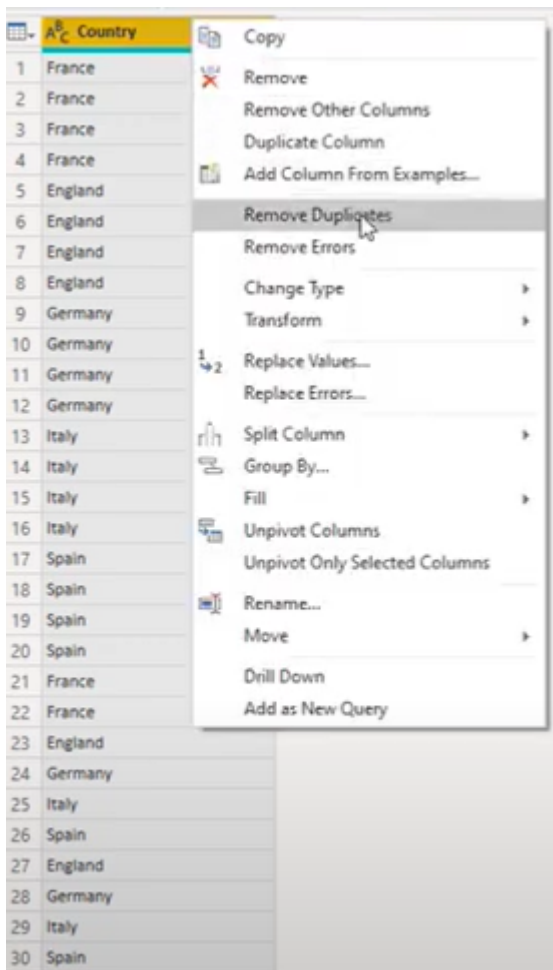


6.

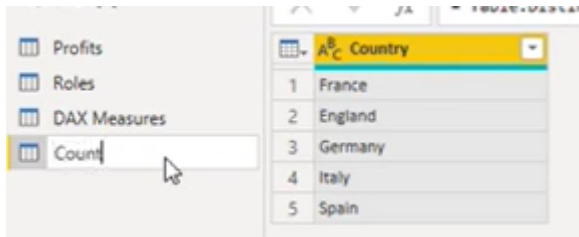


7.

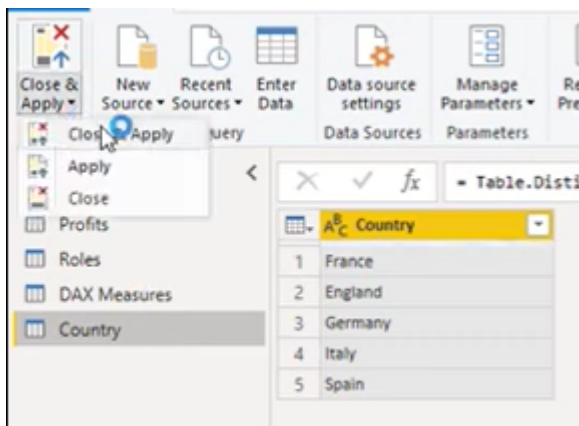
The screenshot shows the 'Country' column selected in the Power Query editor. The context menu is open, displaying various transformation options. The 'Remove Other Columns' option is highlighted, indicating the next step in the data cleaning process.



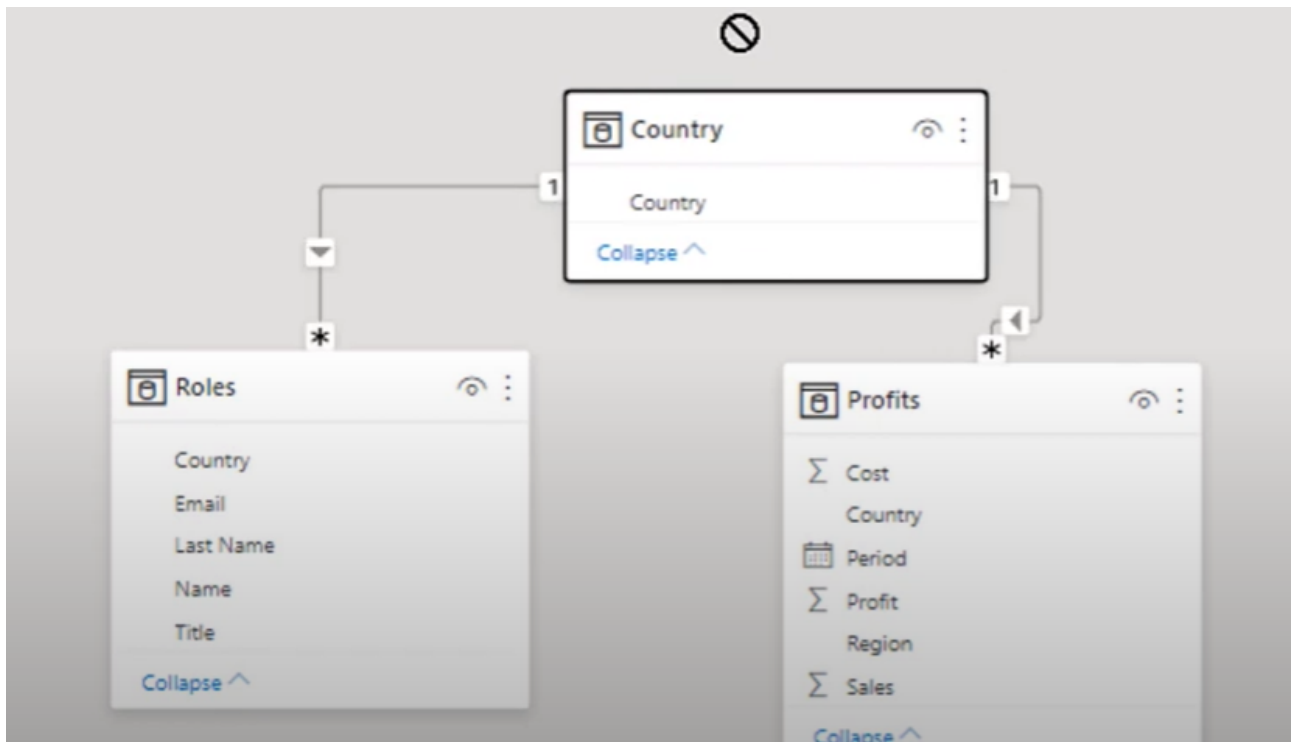
9.



10.



11.



12.

Edit relationship

Select tables and columns that are related.

Roles

Name	Last Name	Title	Country	Email
Jenny	McFarland	Sr Analyst	France	Jenny.McFarland@company.com
Mark	Lopez	Vice President	France	Mark.Lopez@company.com
Mark	Lopez	Vice President	England	Mark.Lopez@company.com

Country

Country
France
England
Germany

Cardinality: Many to one (*:1)

Cross filter direction: Both

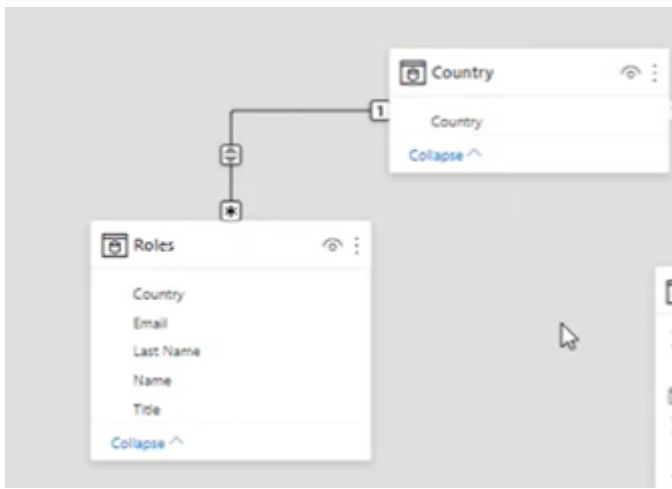
☒ Make this relationship active

☐ Assume referential integrity

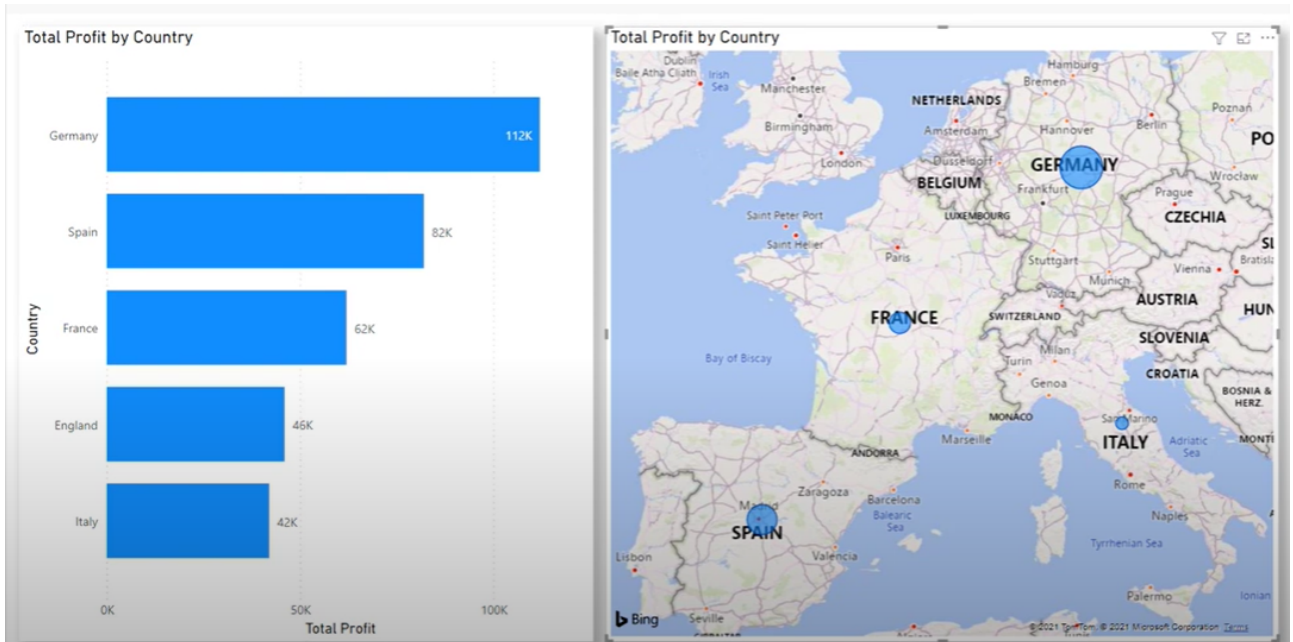
☒ Apply security filter in both directions

OK Cancel

13.



14.



15.

Manage roles

Roles

Dynamic RLS

Create Delete

Tables

Country

DAX Measures

Profits

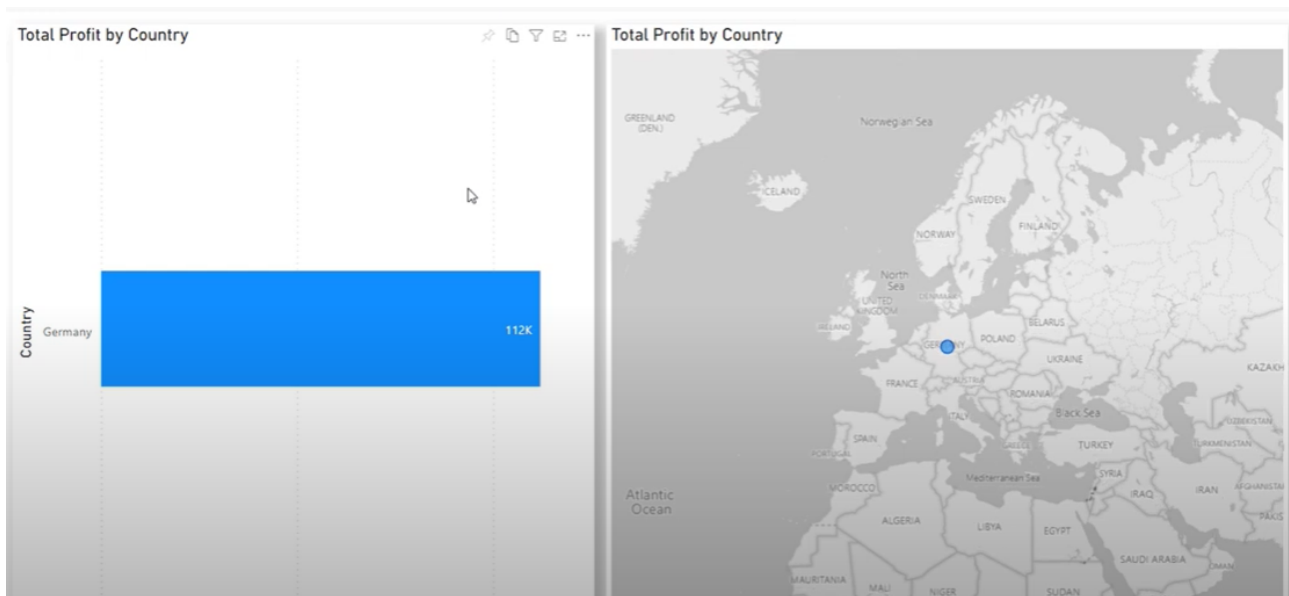
Roles

Table filter DAX expression

[Email] = USERPRINCIPALNAME()

16.

17.



1. Country Relationship as a Bridge

- You created a third table called `Country` that acts as a bridge between the `Roles` and `Profits` tables.
- Both `Roles` and `Profits` share the `Country` column, which connects them via this `Country` table.
- Since the `Roles` and `Country` tables have a **bi-directional relationship**, the filter applied to one table can automatically filter the connected table(s).

2. RLS and Dynamic Security

- You set up **dynamic Row-Level Security (RLS)** by creating a security role with the condition: `[Email] = USERPRINCIPALNAME()`.
- This RLS checks the user's email against the `Roles` table, allowing the report to display only the data where the logged-in user (through Power BI Service) has a corresponding entry in the `Roles` table.

3. Filter Propagation to the Profits Table

- When the report is viewed by a user (e.g., someone from Germany), the RLS filters the `Roles` table to only show rows where `Email = USERPRINCIPALNAME()` and the `Country` matches the user (in your example, "Germany").
- Because of the **bi-directional relationship** between `Roles` and `Country`, this filter is passed to the `Country` table.

- The filter on the `Country` table is then propagated to the `Profits` table, which is also connected by the `Country` column. As a result, only the data for "Germany" is visible in the `Profits` table, even though the RLS is applied directly on the `Roles` table.

Summary of How the Filtering Works:

- The RLS dynamically filters the `Roles` table based on the user's email.
 - The bi-directional relationship between `Roles` and `Country` ensures the filter propagates to the `Country` table.
 - Since the `Profits` table is also connected to the `Country` table, the filter propagates to it as well, limiting the data to only the selected `Country` (Germany, in this case).
-

Summary: RLS in Practice

- **Static RLS:** Manually assign roles and filters.
- **Dynamic RLS:** Use user attributes (like email or ID) to filter data automatically.
- **Testing and Validation:** Always test your RLS implementation to ensure users are only seeing the data relevant to them.

RLS is a powerful feature for organizations with multiple user roles or access levels, ensuring data security while providing relevant insights to each user.