**Task 1**

Step 1: Define Security Groups

I created roles for United States region.

Step 2: Create Security Roles in Power BI Desktop

* Open Power BI Desktop: Open your report in Power BI Desktop.
* Navigate to Modeling Tab: Go to the Modeling tab on the ribbon.
* Manage Roles: Click on Manage Roles to define your security roles.
* Create New Role:
* Click on Create to add a new role.
* You can name the role, for example, Sales, Finance, or HR.
* For role, define a filter on the columns that represent region.
* After creating the role, you can test the role to make sure the filter works as expected.
* Click View as Roles, and select the role you want to test. This will show the report as if you are that specific role.

Step 3: Publish to Power BI Service

* Click Publish in Power BI Desktop.
* Choose the workspace where you want to publish the report.
* Configure RLS in Power BI Online Service:
* After publishing, go to Power BI Online (Service).
* Open the report and navigate to Security in the dataset settings.

Step 4: Configure RLS in Power BI Service

Go to Dataset Settings:

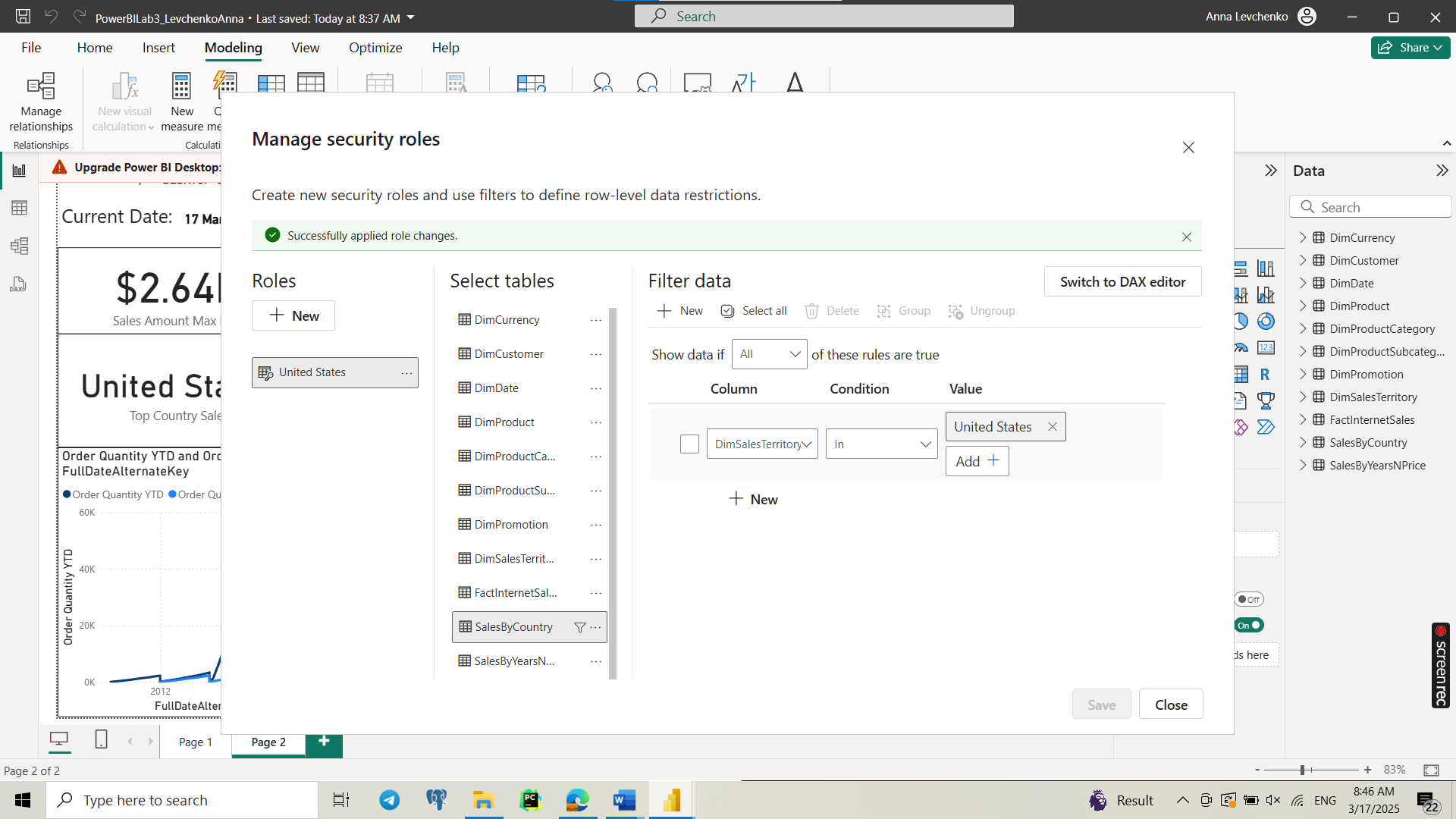
* In Power BI Service, go to Workspace > Datasets.
* Find the dataset associated with your report.
* Manage Roles in Power BI Service:
* Click on the More Options (three dots) next to the dataset.
* Select Security.

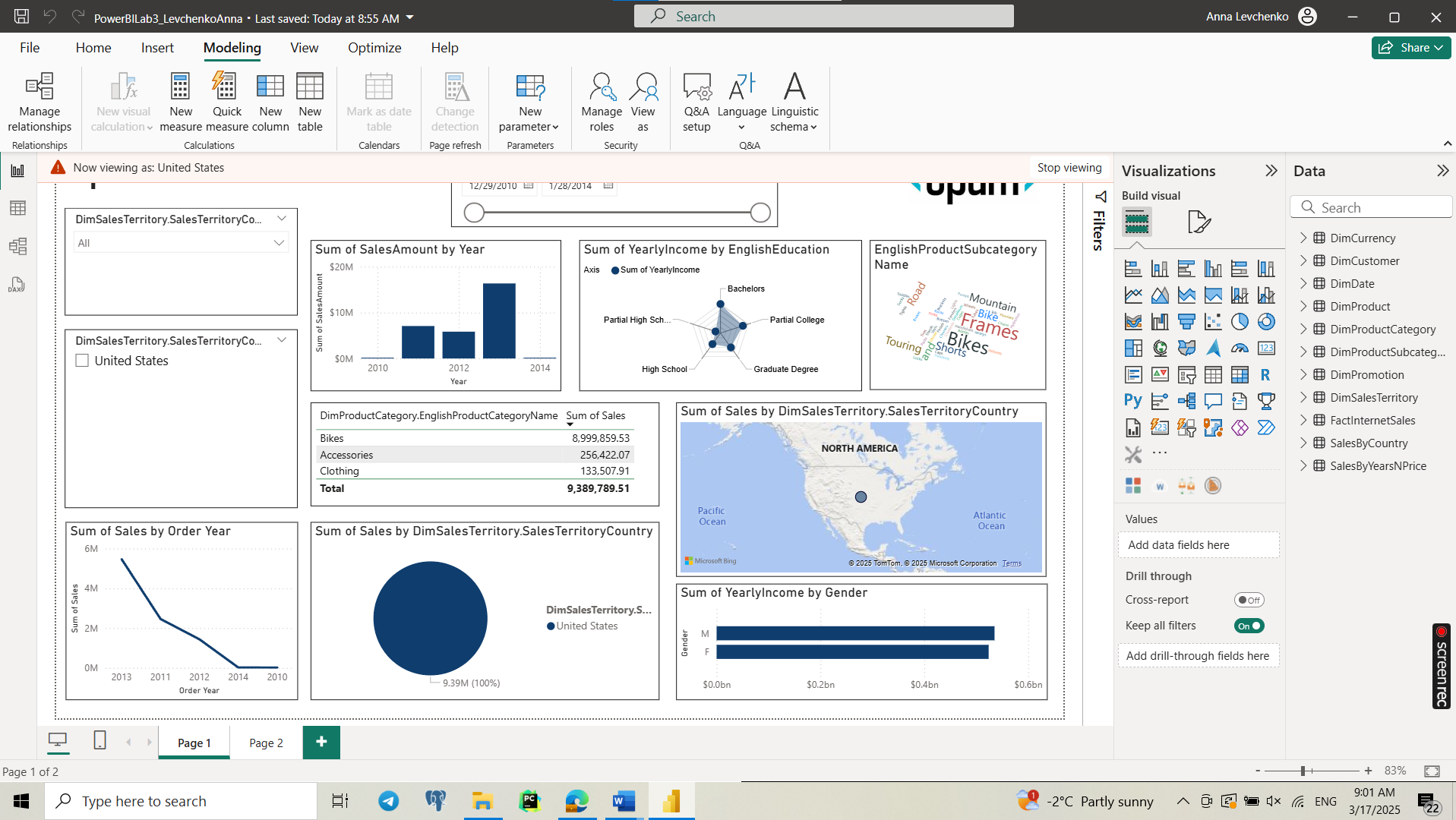
Assign Users to Roles:

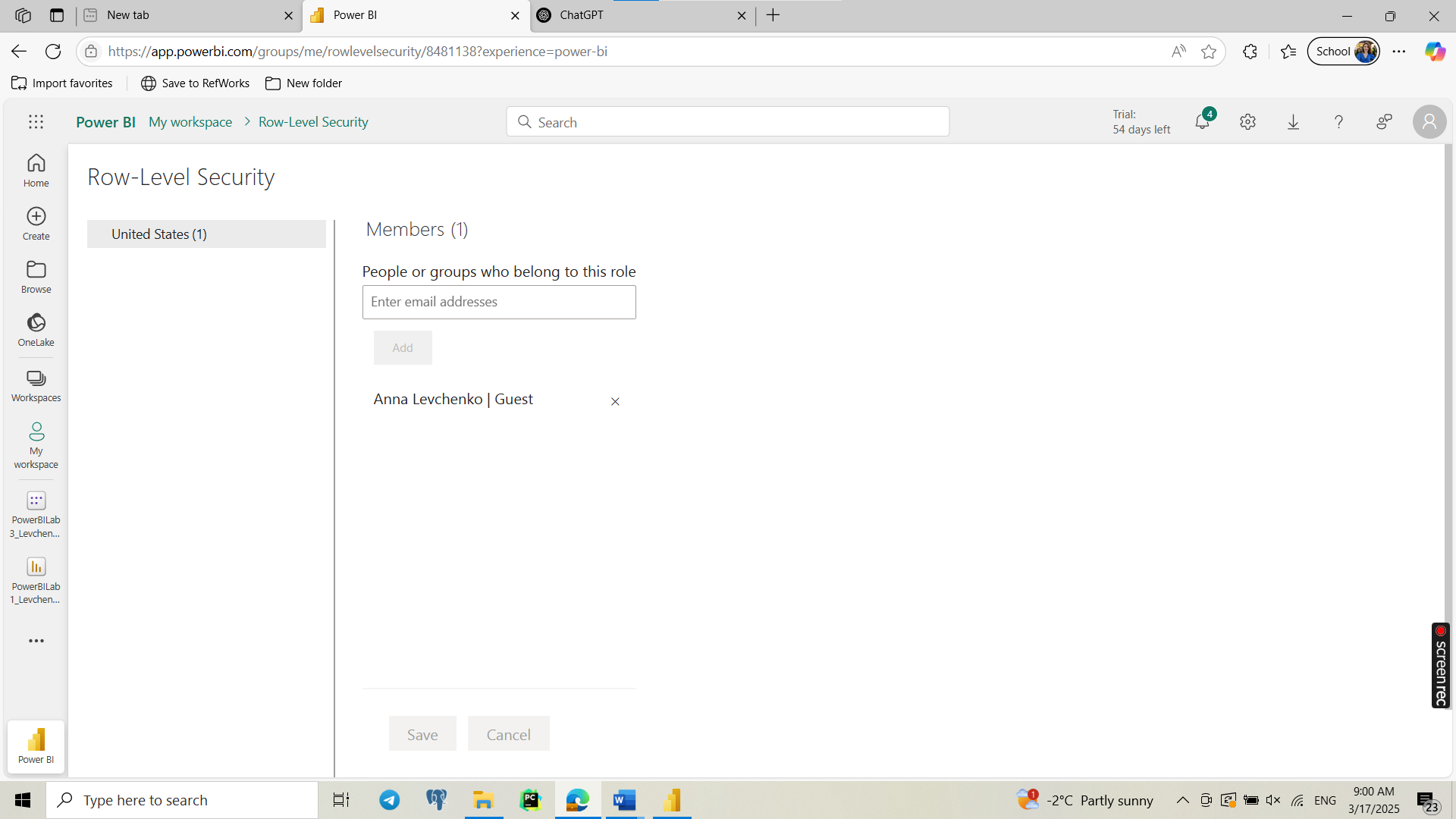
* Under Security, you’ll see the roles you created in Power BI Desktop.
* For each role, click Add to assign users or groups who should belong to that role.
* You can assign users by their email addresses.

Save the Configuration:

* Once the users are assigned to their respective roles, click Save.







**Task 2**

1. Create the Bar Chart and Map Visual

* Insert a Bar Chart and a Map Visual in the same position on the report page.
* Ensure both visuals are fully overlapping to create a seamless toggle effect.

2. Use Selection Pane to Manage Visibility

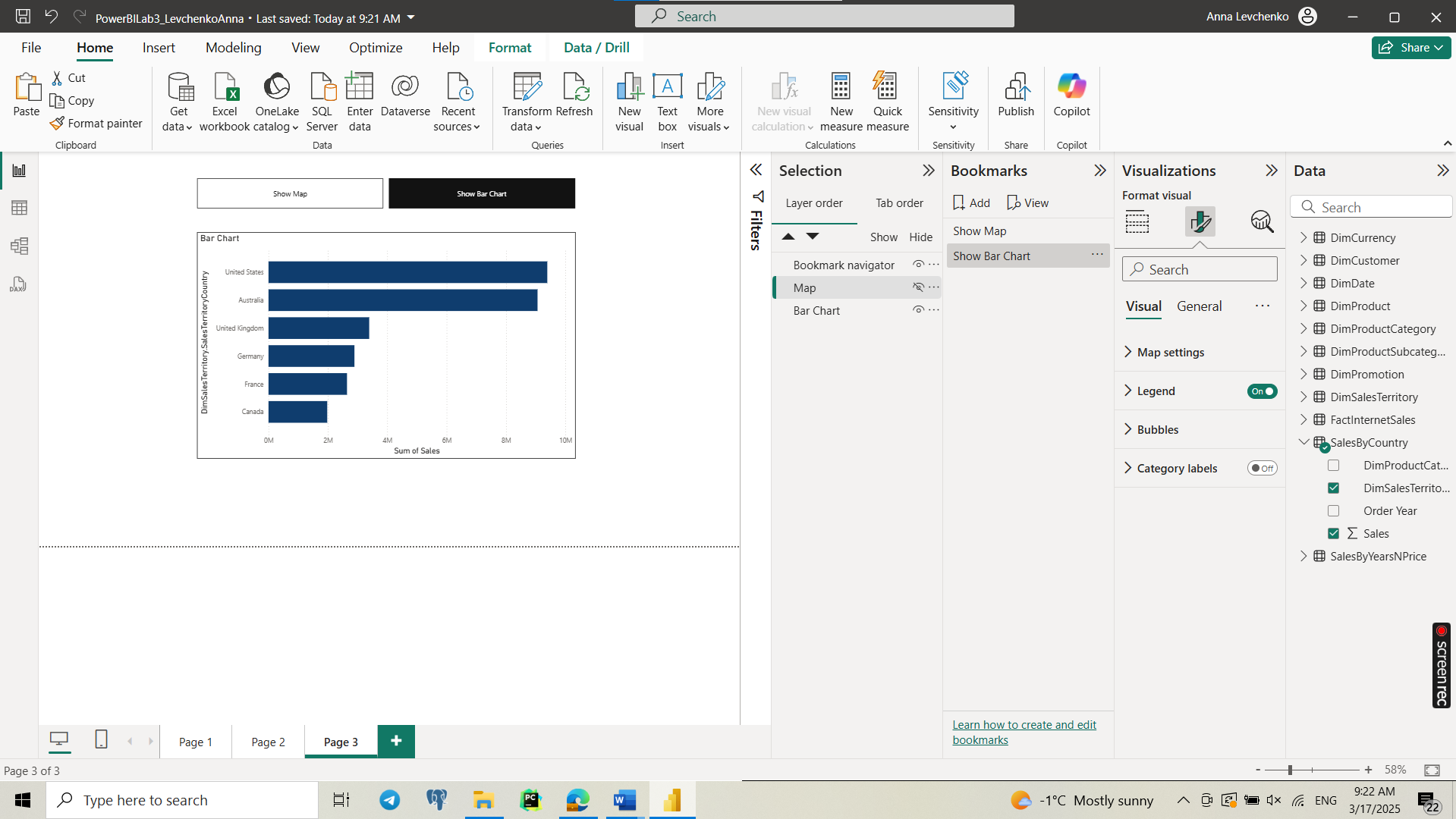
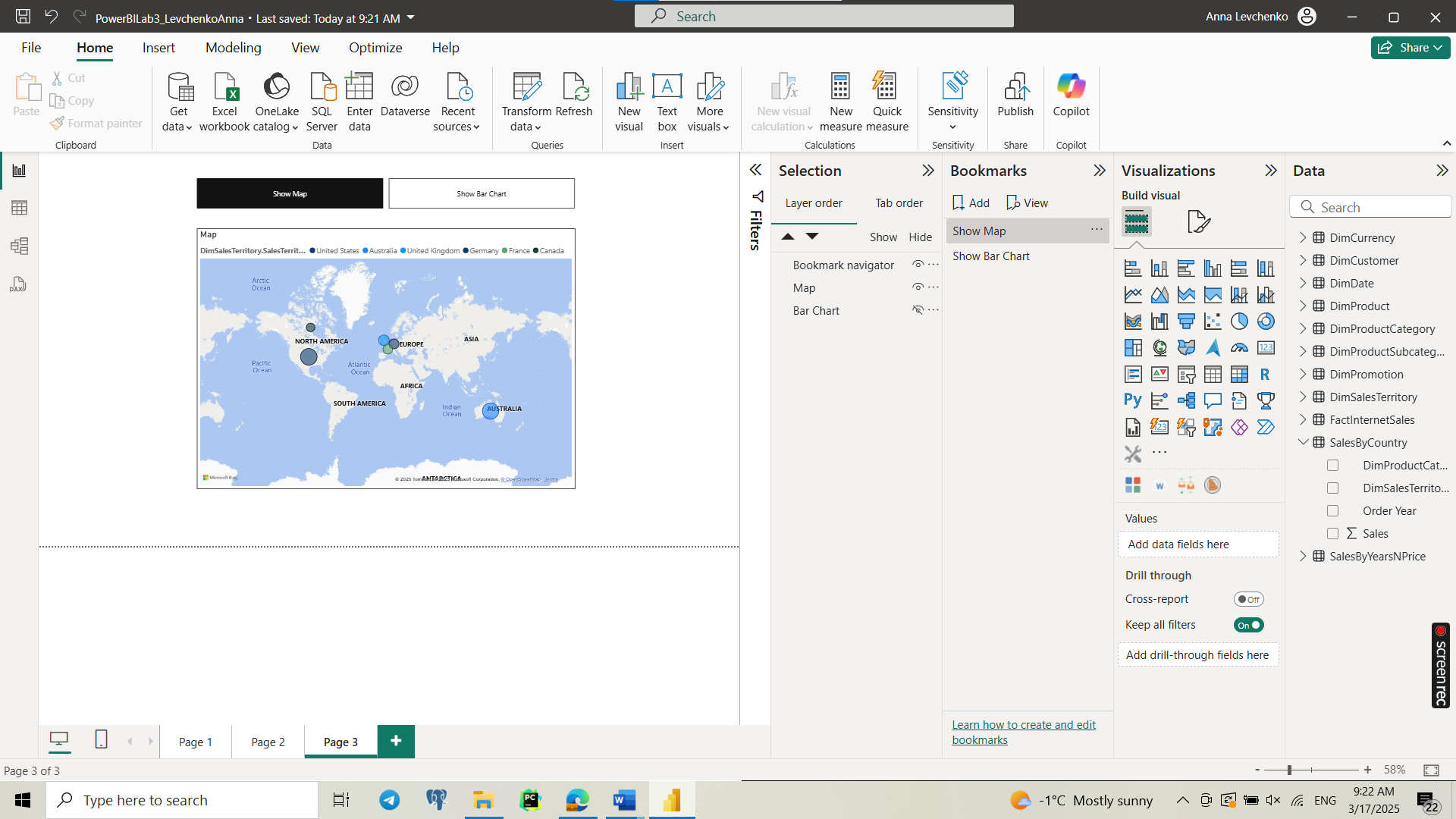
* Go to View → Enable Selection Pane.
* Rename the visuals clearly (e.g., "Bar Chart" and "Map").

3. Create Bookmarks

* Open the Bookmarks Pane and add two bookmarks:
* "Show Bar Chart" (where only the Bar Chart is visible, and the Map is hidden).
* "Show Map" (where only the Map is visible, and the Bar Chart is hidden).
* In each bookmark, update and uncheck "Data" (to prevent affecting filters).

4. Add Buttons to Toggle Views

* Insert button (from the Insert menu) and position it strategically:
* “View Bar Chart” → Assign it to the Show Bar Chart bookmark.
* “View Map” → Assign it to the Show Map bookmark.



**Task 3**

Step 1: Create a Selection Table (Disconnected Table)

We need a table that contains the options MTD, QTD, and YTD. You can create this in Power BI Desktop using DAX:

TimePeriodSelection =

DATATABLE(

"Period", STRING,

{

{"MTD"},

{"QTD"},

{"YTD"}

}

)

* This table is not connected to other tables.
* It serves as the slicer for user selection.

Step 2: Create a Dynamic Measure

Modify your current Sales Amount YTD measure to dynamically switch between MTD, QTD, and YTD:

Sales Amount Selected Period =

VAR SelectedPeriod = SELECTEDVALUE(TimePeriodSelection[Period], "YTD")

RETURN

SWITCH(

SelectedPeriod,

"MTD", CALCULATE(

SUM(FactInternetSales[SalesAmount]),

DATESMTD(DimDate[FullDateAlternateKey])

),

"QTD", CALCULATE(

SUM(FactInternetSales[SalesAmount]),

DATESQTD(DimDate[FullDateAlternateKey])

),

"YTD", CALCULATE(

SUM(FactInternetSales[SalesAmount]),

DATESYTD(DimDate[FullDateAlternateKey])

),

CALCULATE( // Default to YTD if nothing is selected

SUM(FactInternetSales[SalesAmount]),

DATESYTD(DimDate[FullDateAlternateKey])

)

)

Explanation:

* SELECTEDVALUE(TimePeriodSelection[Period], "YTD")
  + Gets the user's selection from the slicer.
  + Defaults to "YTD" if nothing is selected.
* SWITCH(...)
  + Dynamically switches between MTD, QTD, and YTD based on selection.
  + Uses DATESMTD(), DATESQTD(), and DATESYTD() to calculate the respective periods.

Modify your Order Quantity YTD measure to dynamically switch between MTD, QTD, and YTD:

Order Quantity Selected Period =

VAR SelectedPeriod = SELECTEDVALUE(TimePeriodSelection[Period], "YTD")

RETURN

SWITCH(

SelectedPeriod,

"MTD", CALCULATE(

SUM(FactInternetSales[OrderQuantity]),

DATESMTD(DimDate[FullDateAlternateKey])

),

"QTD", CALCULATE(

SUM(FactInternetSales[OrderQuantity]),

DATESQTD(DimDate[FullDateAlternateKey])

),

"YTD", CALCULATE(

SUM(FactInternetSales[OrderQuantity]),

DATESYTD(DimDate[FullDateAlternateKey])

),

CALCULATE( // Default to YTD if nothing is selected

SUM(FactInternetSales[OrderQuantity]),

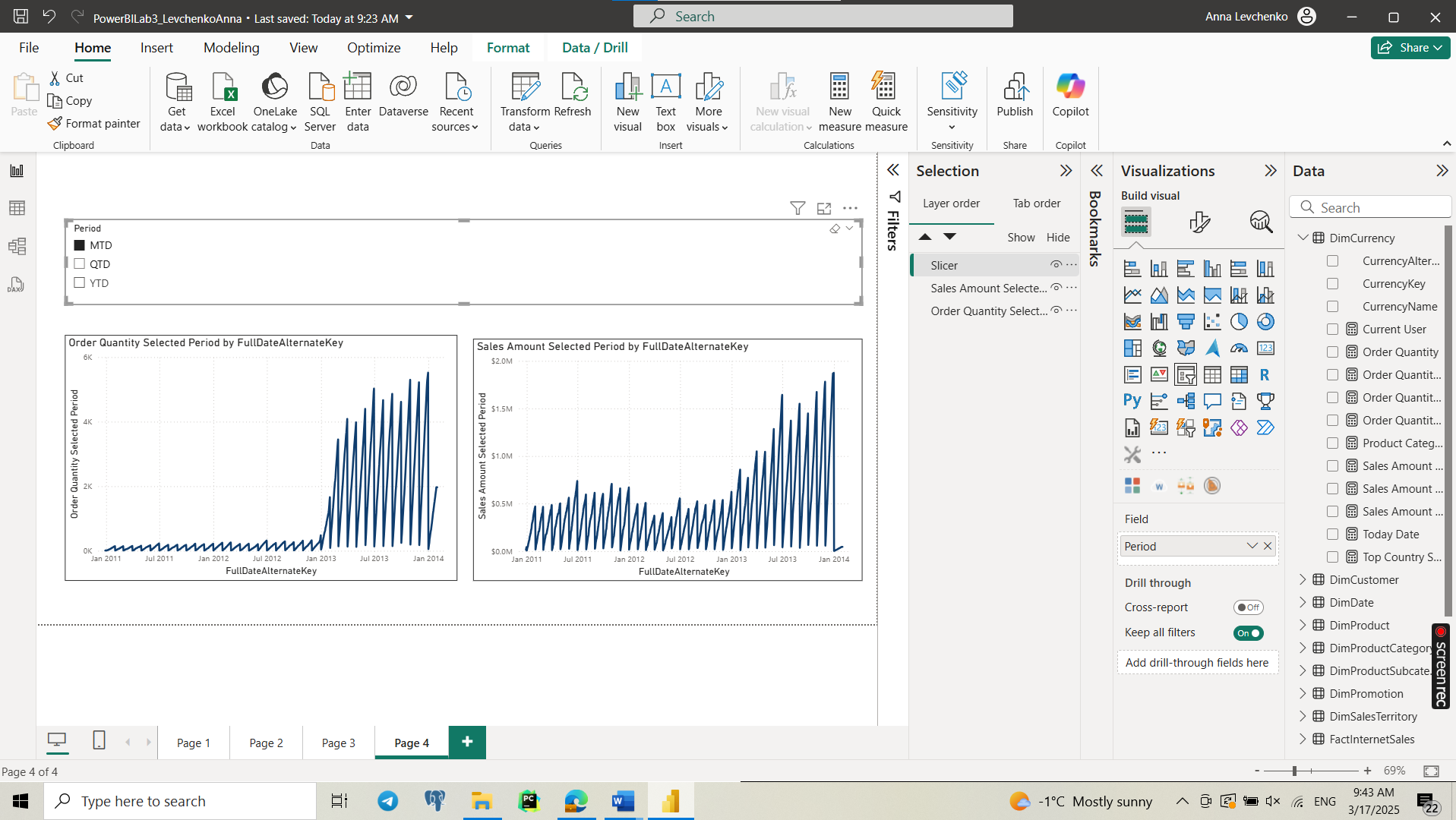
DATESYTD(DimDate[FullDateAlternateKey])

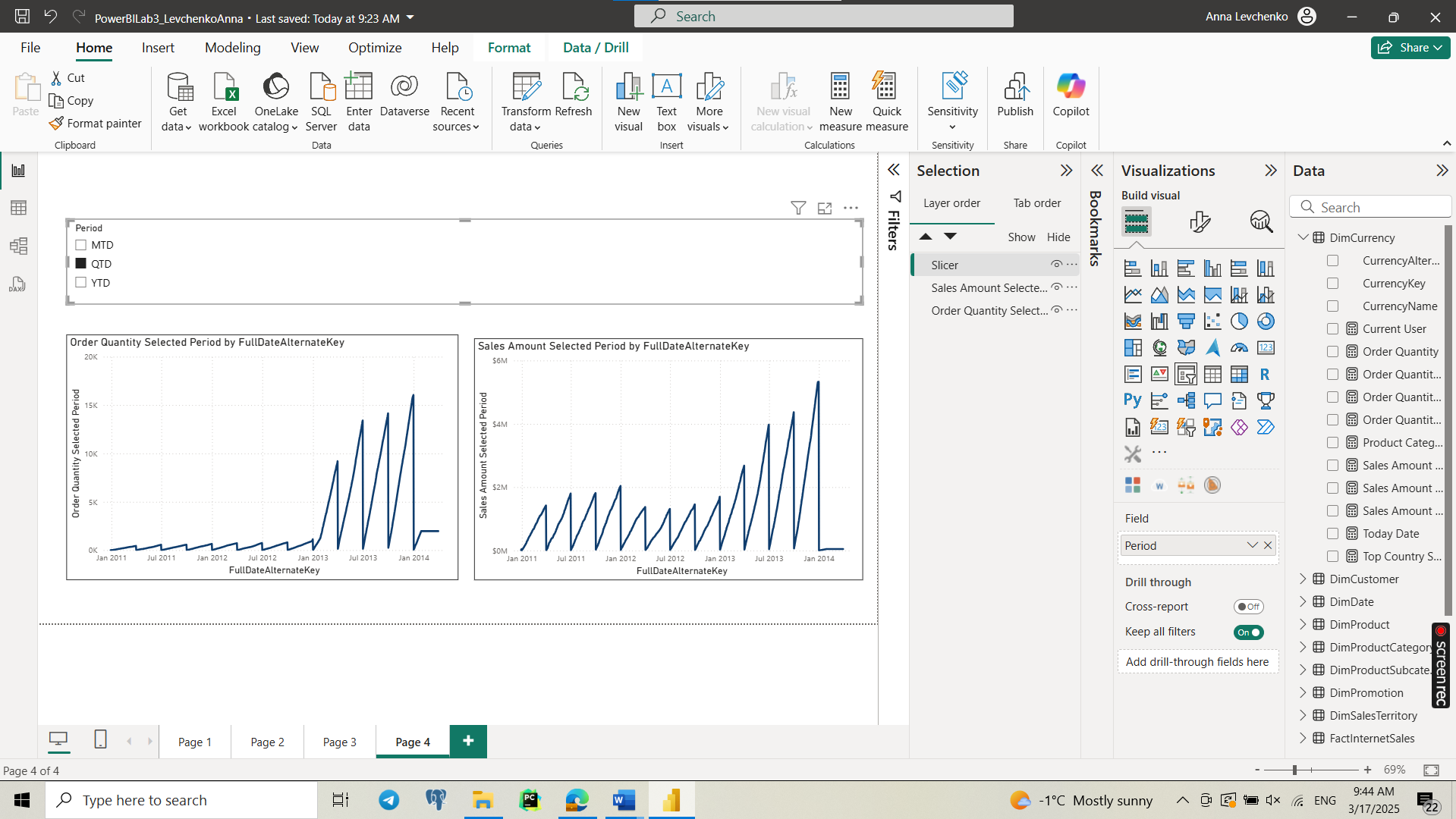
)

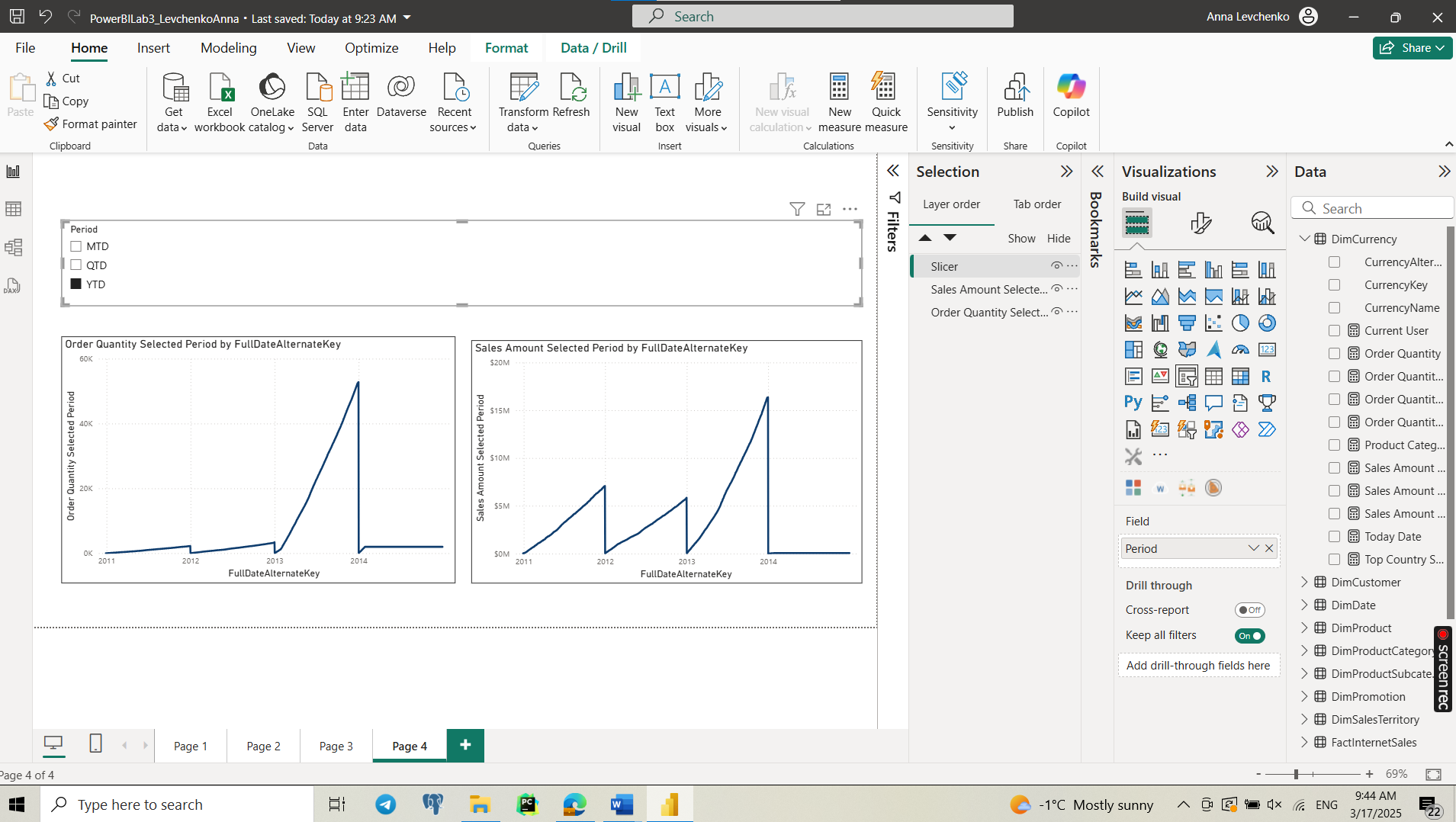
)

Step 3: Add a Slicer to the Report

1. Add a Slicer visual to the report.
2. Use the column "Period" from TimePeriodSelection as the slicer field.
3. Now users can choose between MTD, QTD, and YTD.







**Task 4**

1. Select the Measure to Analyze

For this decomposition tree, I used Sum of Sales from the FactInternetSales table:

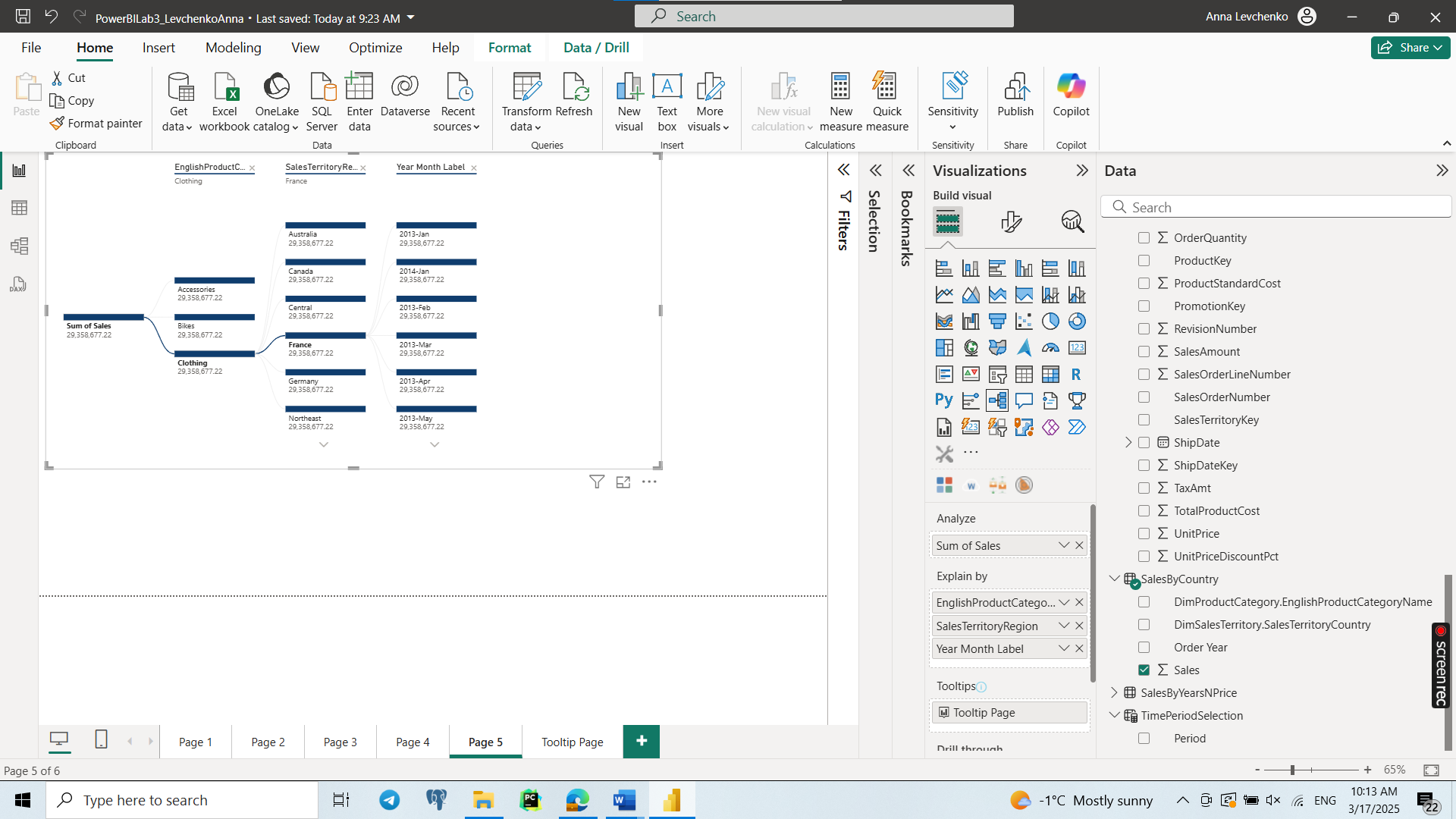
2. Insert a Decomposition Tree Visual

* Go to Visualizations → Select Decomposition Tree.
* Drag Total Sales to the Analyze field.

3. Add Independent Variables

To analyze sales breakdown, I added three independent variables in the Explain By section:

* Product Category (from DimProductCategory)
* Customer Region (from DimCustomer)
* Order Year (from DimDate)



**Task 5**

1. Create a New Tooltip Page

* Click New Page at the bottom of Power BI.
* Go to Format Pane → Page Size → Set to Tooltip.
* Set Page View to Fit to Page.
* Rename the page to something like "Tooltip Page".

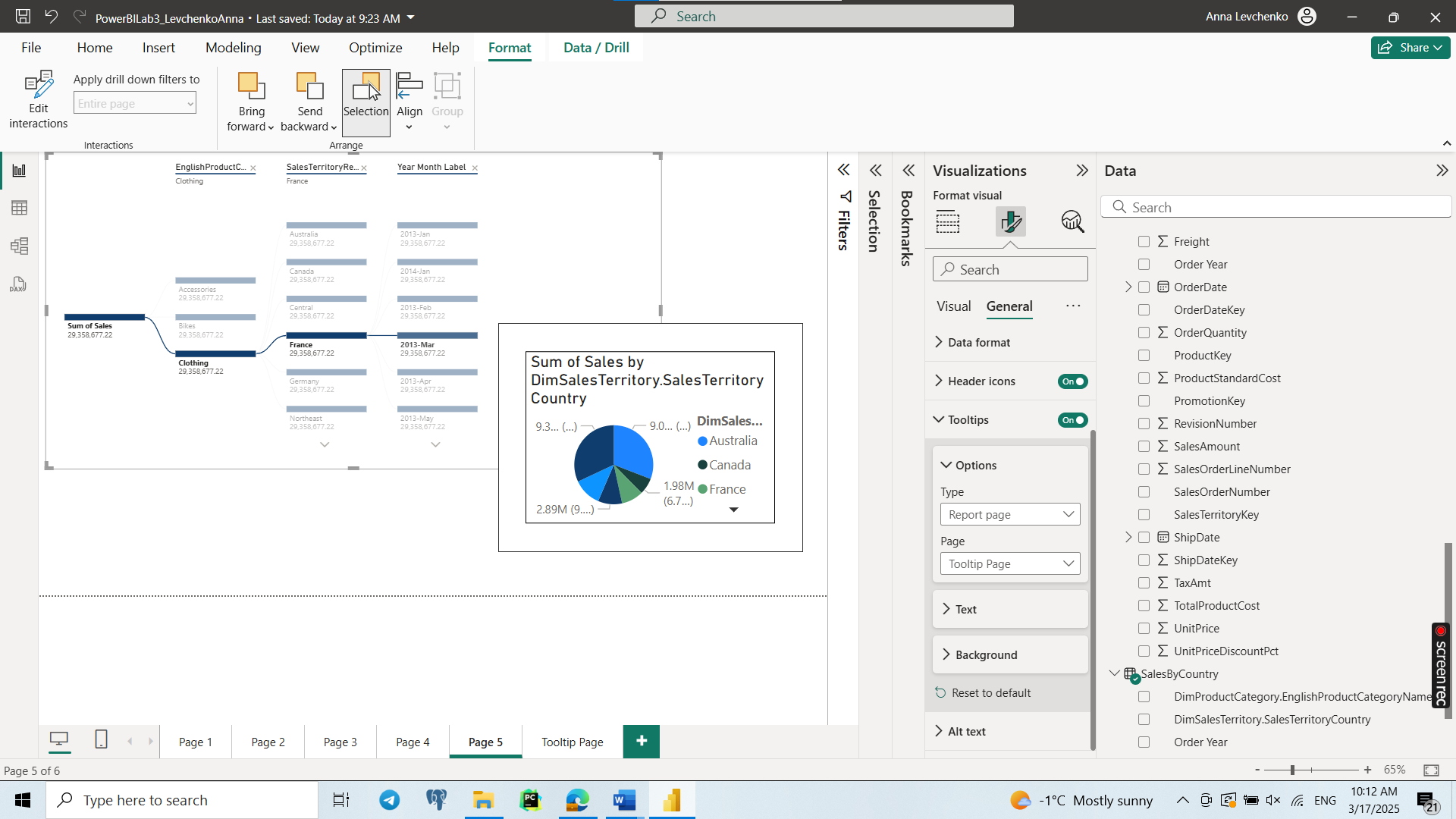
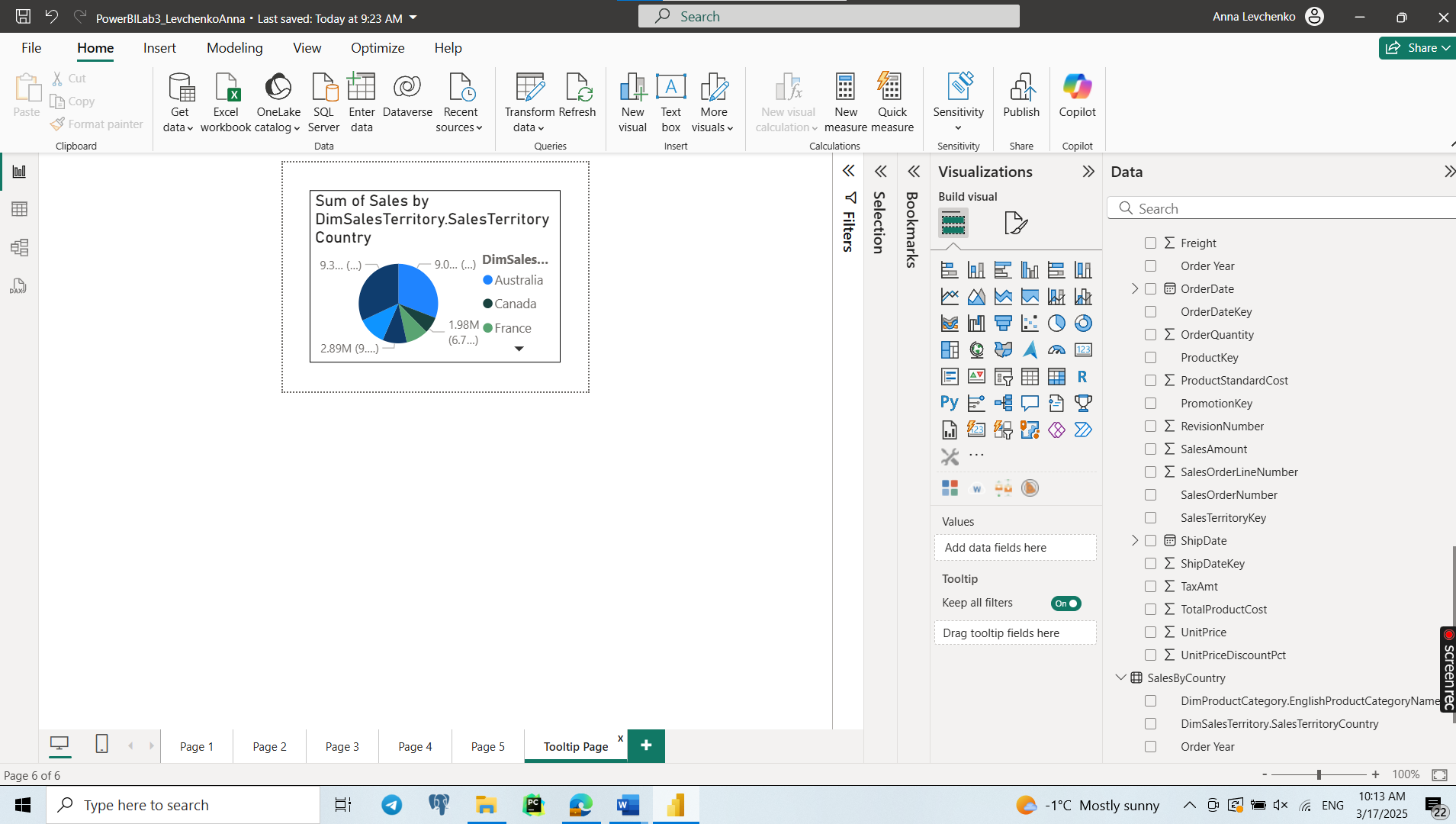
2. Add Visuals to the Tooltip Page

* Insert a Chart to display additional insights.
* Use the same filtering context as your main visual.

3. Apply the Tooltip to Your Main Visual

* Go back to the main report.
* Select the visual → Open Format Pane.
* Expand Tooltip → Change Type to Report Page.
* Choose "Tooltip Page" from the dropdown.

Now, when users hover over the main visual, they will see a custom tooltip with charts and extra data instead of just text fields.



**Task 6**

Step-by-Step Process to Create the Custom Visual:

1. Dataset Grouping: The dataset will be grouped by Product Category, and we'll calculate the sum of Sales for each category.
2. Dynamic Coloring: The bars will be colored based on the sales values, where higher sales will have a different color (e.g., green for high sales and red for low sales).
3. Text Annotations: We'll add text annotations to display the sales value on top of each bar.
4. Interactive Elements: Adding a feature that makes the bars change colors dynamically based on the sales value.

Opportunities in Comparison with Default Power BI Visuals:

* Color-Coding for Sales: Default bar charts in Power BI do not have dynamic color-coding based on sales values (except for manual conditional formatting). This custom visual automates the process of color-coding bars based on the sales range, which makes the chart easier to read and more insightful.
* Customizable Annotations: Default visuals don't automatically display values above the bars in all cases. By using this Python custom visual, you can add dynamic annotations that appear directly on the chart, providing more context.

