**Exercise: Improving DAX performance**

**Introduction**

You have now gained a clearer understanding of Power BI and grasped the significance of DAX variables in optimizing your report performance. Through this exploration, you've discovered how variables can be instrumental in troubleshooting and relieving performance challenges within your reports and visualizations. As you continue this journey, the potential of these variables to streamline complex calculations and bolster report performance will become even more apparent.

In this exercise, you will need to apply your knowledge to identify DAX performance issues and implement optimization in an Adventure Work Microsoft Power BI report. Your task is to:

* Use the **Table view** to examine the data in the report
* Use the **Performance Analyzer** to identify slow visuals and DAX formulas
* Test the changes with the **Performance Analyzer** to ensure improved report efficiency

**Case study**

Imagine you're a data analyst at Adventure Works. You've just built an impressive dashboard that showcases sales data over time, integrating data from across the globe and telling a comprehensive story of the company's sales operations. There's just one small problem–the masterpiece is in slow motion. Whenever you try to refresh or manipulate the data, it takes forever. Even a few seconds can feel like an eternity in today's fast-paced business landscape. Every extra second it takes to load is a dent in its efficiency. The culprit? It's a DAX query that's dragging everything down.

**Instructions**

Download the Adventure Works Power BI report, *Improving DAX performance*, and follow the steps below to complete the exercise.

[Improving DAX performance](https://d3c33hcgiwev3.cloudfront.net/ps6lxhzzSuKV5WzpeR70jw_f864a2983256422ab61d9813246f9de1_Improving-DAX-performance.pbix?Expires=1711929600&Signature=G-CA0OkzH4KzwpTL9ME1LIPZfBDf3hii-dgCiW5xQFi0gkdJdxvOgxtzvAPPtIZJ-HKmDMOnYBZs1mLqdgIcp6wSWoUNL5swNpuL7l8MEH9~XCTlrx~PJR0gWfyi515h5M2xccVt6hukwLq6hRUYrHAzMWaZJyEV-H2sShXP9yY_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A" \t "_blank)

[PBIX File](https://d3c33hcgiwev3.cloudfront.net/ps6lxhzzSuKV5WzpeR70jw_f864a2983256422ab61d9813246f9de1_Improving-DAX-performance.pbix?Expires=1711929600&Signature=G-CA0OkzH4KzwpTL9ME1LIPZfBDf3hii-dgCiW5xQFi0gkdJdxvOgxtzvAPPtIZJ-HKmDMOnYBZs1mLqdgIcp6wSWoUNL5swNpuL7l8MEH9~XCTlrx~PJR0gWfyi515h5M2xccVt6hukwLq6hRUYrHAzMWaZJyEV-H2sShXP9yY_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A" \t "_blank)

**Step 1: Open your project**

1. Once you have Power BI Desktop open, in the top left corner, select the **File** menu.
2. Navigate through this to the location where the *Improving DAX performance.pbix* report file is saved.
3. Select the file and select **Open** in the file explorer window.

**Step 2: Explore the data in Table view**

1. When you select the **Table view** icon, Power BI takes you to the **Table view**, which allows you to explore the data contained in your project.
2. Select the **Sales** dataset on the right of the screen and take a moment to observe the first ten records.
3. Note which record has the highest **Product Weight** value.
4. Note the **Day of Week** with the highest frequency of sales records.
5. Finally, use the icons on the vertical toolbar on the left side of the Power BI interface to switch back to the **Report view**.

**Step 3: Access the Performance Analyzer**

1. Once you're in the **Report view**, you'll first need to open the **Performance Analyzer**. Locate and select the View tab on the ribbon interface at the top of your Power BI report.
2. Within the **View** tab, find and select the **Performance Analyzer** option.
3. Within the **Performance Analyzer** pane, locate and select the **Start Recording** button.

**Step 4: Refresh the report**

* Now, it's time to refresh your report. You can accomplish this in two ways: either by selecting the **Refresh** button situated in the **Home** tab of the ribbon interface or by directly interacting with the report.

**Step 5: Observe the results**

1. Once the report has finished refreshing, review the **Performance Analyzer** pane. Pay special attention to any visual items that take significantly longer to load than others.
2. For these slower visuals, drill down into the details by selecting the **+** symbol beside the visual item's name.
3. You should find that the **Total Sales by Year, Quarter, Month and Day** area chart is slowing down the report performance.

**Step 6: Improve DAX performance**

In DAX, the **CROSSJOIN** function generates a cartesian product by pairing every row of the first table with every row of the second table. This operation is conducted regardless of any matching column values between the tables. It merges each row from one table with all the rows from the other table.

1. Locate the **Total Sales** field of the area chart from the **Table** view on your right and select it to view the underlying DAX formula.
2. To simplify the DAX formula, eliminate the nested **CROSSJOIN** and **GENERATESERIES** functions.
3. Instead, use the **SUMX** function enclosed with a single **CROSSJOIN**:

1

2

3

4

5

Total Sales =

SUMX(

CROSSJOIN(VALUES(Sales[Product Category]), VALUES(Sales[Payment Method])),

CALCULATE(SUM(Sales[Sales Total]))

)

**Step 7: Test and review**

* Finally, re-run the performance analyzer to test if the optimization was successful.

**Conclusion**

Think of every data report as a story—a narrative waiting to be told. When there's a delay, the essence of the story gets lost. But when data flows smoothly, it speaks volumes, narrating tales of opportunities, challenges, and growth. Every time you refine a DAX calculation and streamline a query, you're not just improving a report—you're elevating the entire data-driven decision-making process. And in a world increasingly powered by data, that's not just a skill; it's a superpower.

**Exemplar: Improving DAX performance**

**Introduction**

In the exercise *Improving DAX performance*, you were immersed in the world of Adventure Works, the multinational bicycle and accessories powerhouse, where you confronted a unique challenge: a slow Microsoft Power BI report. The mission in hand was to locate and optimize a DAX query that hindered the report’s performance, ensuring that data-driven decisions could be made efficiently and without delay.

More specifically, you were required to:

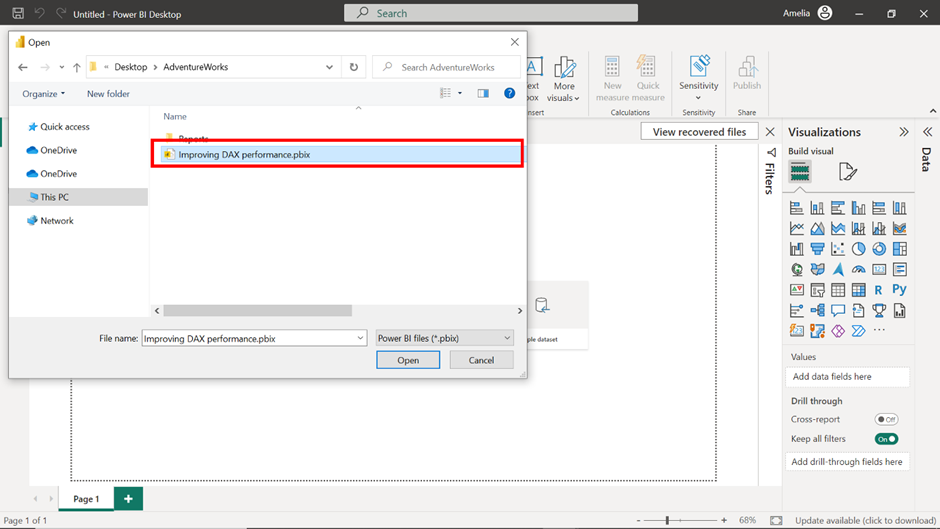
* Download the Adventure Works Power BI report named *Improving DAX performance.pbix* and open it in Power BI Desktop, setting the groundwork for your journey.
* Dive into the **Table view** to familiarize yourself with the dataset available in the report, laying the foundation for understanding the details of the data at hand.
* Harness the power of the **Performance Analyzer** to pinpoint the underperforming visualization and DAX formula in your report.
* Refine the DAX formula of the **Total Sales** field to optimize its performance, replacing the nested functions with a more streamlined version.
* Validate the modifications by re-running the **Performance Analyzer**, ensuring that the tweaks made indeed have a positive effect on the dashboard's efficiency.

This reading serves as a structured walkthrough, guiding you step by step, ensuring that you’re on the right track, and assisting you in comparing your efforts against a standard solution.

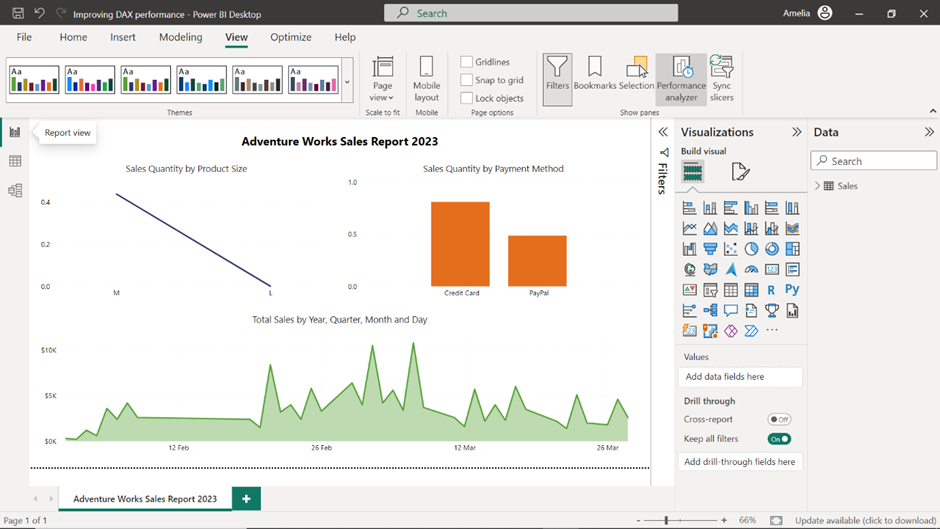
**Improving DAX performance**

**Step 1: Open your project**

1. Once you have Power BI Desktop open, in the top left corner, select the **File** menu.
2. Navigate through this to the location where your *Improving DAX performance* report file is saved.

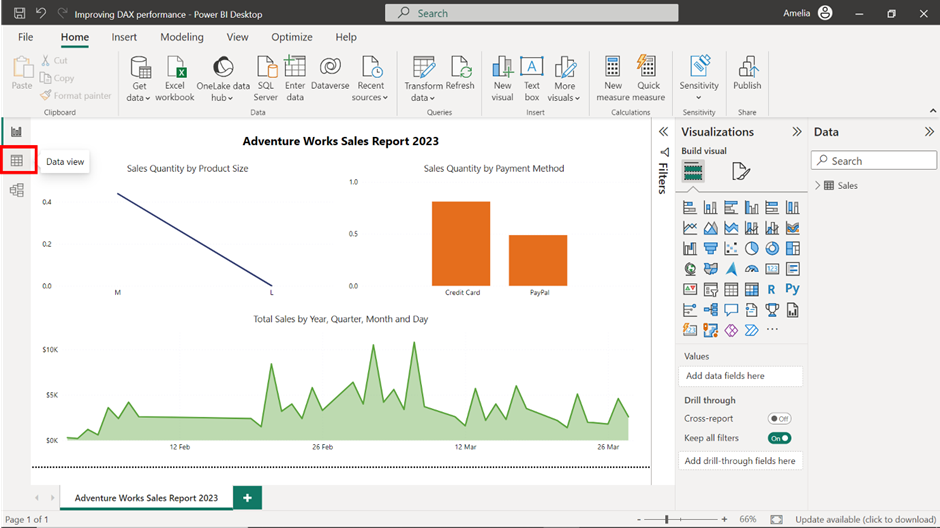


3. Select the file and select **Open** in the file explorer window. This action opens the saved project in the Power BI Desktop application.



**Step 2: Explore the data in Table view**

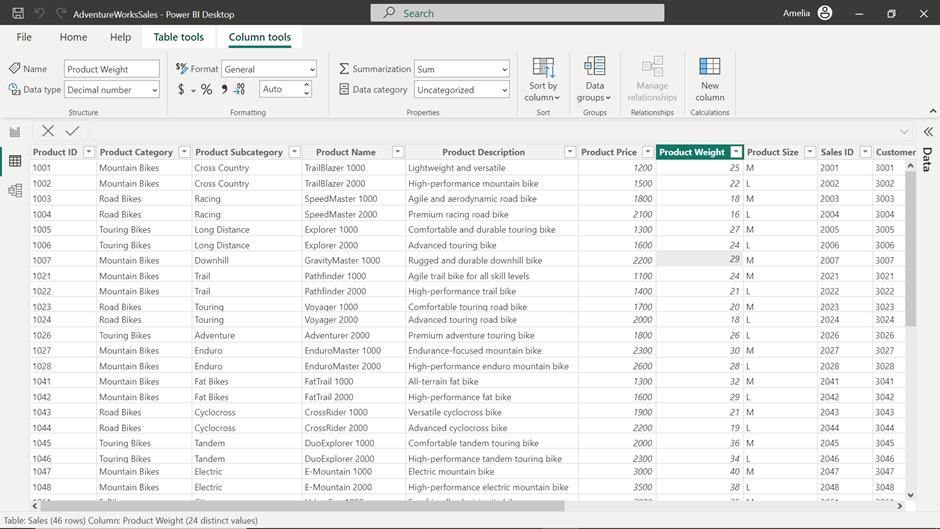
1. Once your report is loaded, on the left side of the Power BI interface, you'll find a vertical toolbar with different icons. The second icon from the top that resembles a table is the **Table view** icon. When you select this, Power BI takes you to the **Table view** which allows you to explore the data contained in your project.



2. Select the **Sales** dataset on the right of the screen and take a moment to observe the first 10 records.

3. Note which record has the highest **Product Weight** value. With a weight of 29 units, the **GravityMaster 1000** stands as the heaviest bicycle.

4. Then, note the **Day of Week** that has the highest frequency of sales. Monday stands out as the busiest day at Adventure Works–with three sales records, it has the highest frequency in the **Day of Week** column.



5. Finally, use the icons on the vertical toolbar on the left side of the Power BI interface to switch back to the **Report view**.

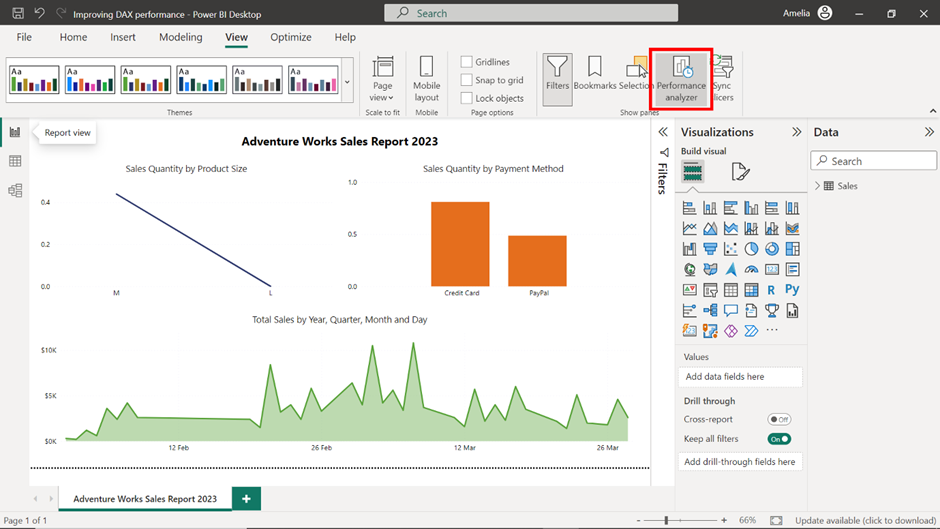
A screenshot of a computer

Description automatically generated

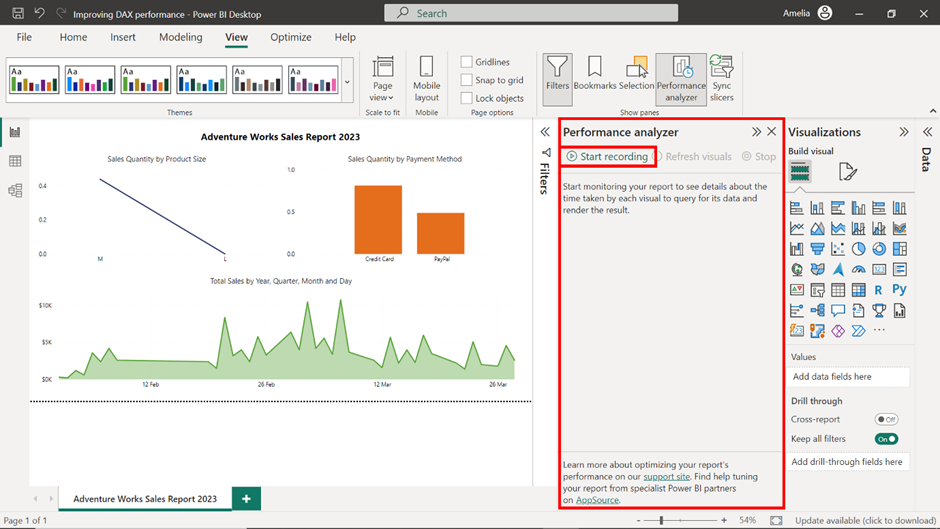
**Step 3: Access the Performance Analyzer**

1. Once you're in the **Report view**, you first need to open the **Performance Analyzer**. On the ribbon interface at the top of your Power BI report, locate and select the **View** tab.

2. Within the **View** tab, find and select the **Performance Analyzer** option.

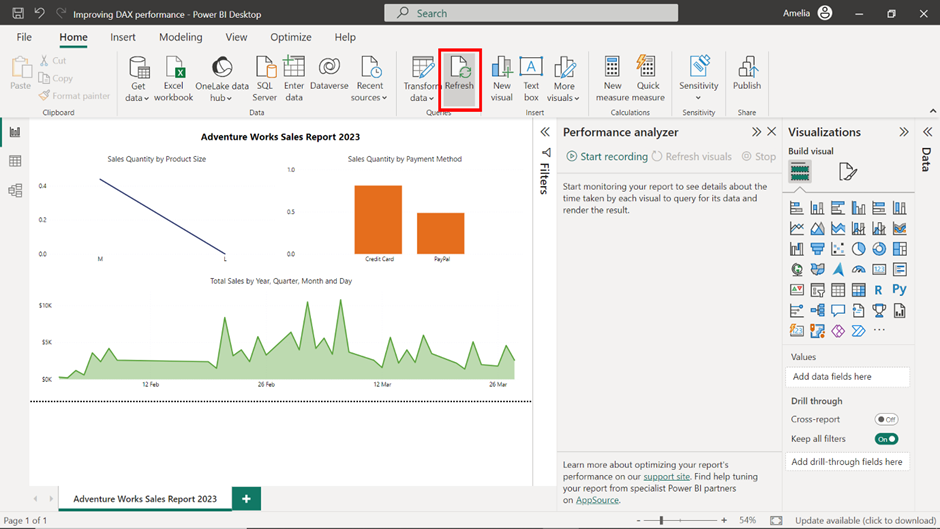


3. Upon selection, you’ll notice a pane unfold on the right side of the Power BI window. This **Performance Analyzer** pane is designed to showcase real-time performance metrics of your report visuals. Within the **Performance Analyzer** pane, locate and select the **Start Recording** button.



**Step 4: Refresh the report**

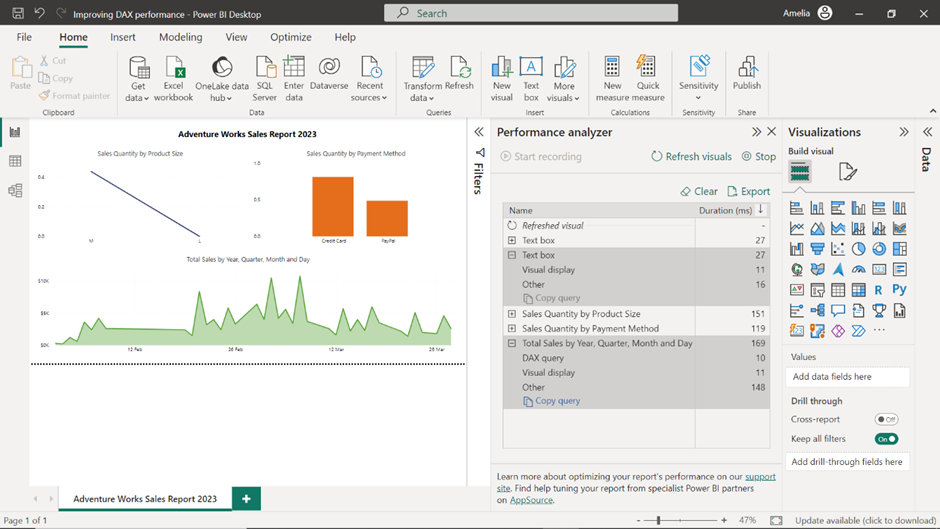
* Now, it's time to refresh your report. You can accomplish this in two ways: either by selecting the **Refresh** button situated in the **Home** tab of the ribbon interface or by directly interacting with the report.



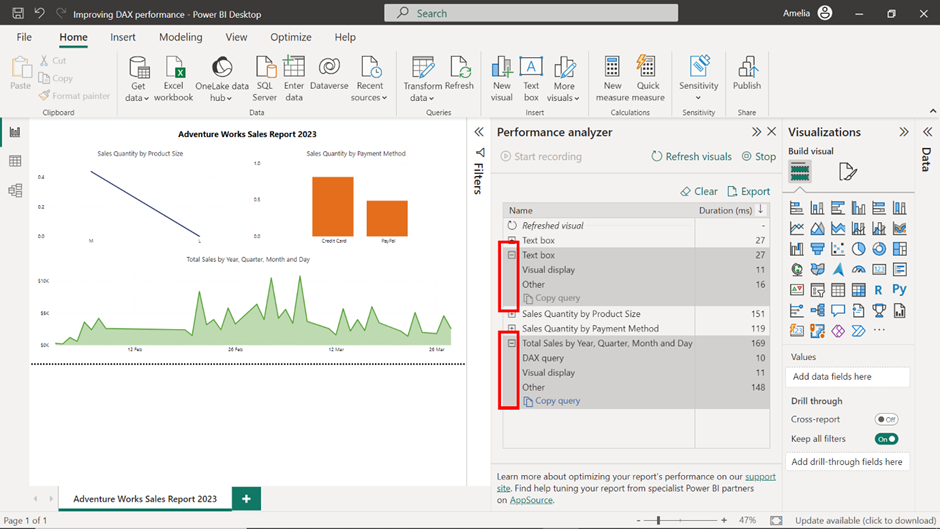
**Note:** As you interact with the report while the **Performance Analyzer** is recording, it will track and document the time taken to load each individual visual item. This data will be crucial for diagnosing performance issues.

**Step 5: Observe the results**

1. Once the report has finished refreshing, review the **Performance Analyzer** pane. A list of all the visual items in your report and their respective load times is available. Pay special attention to any visual items that take a significantly longer time to load compared to others. This ensures that you're not wasting time optimizing calculations that are already performing well.



2. For these slower visuals, drill down into the details by selecting the **+** symbol beside the visual item’s name. This will provide a detailed breakdown of the DAX query time and the visual rendering time, helping you understand where the bottleneck lies. If the DAX query time is high, then your efforts should be directed towards optimizing the DAX measures.

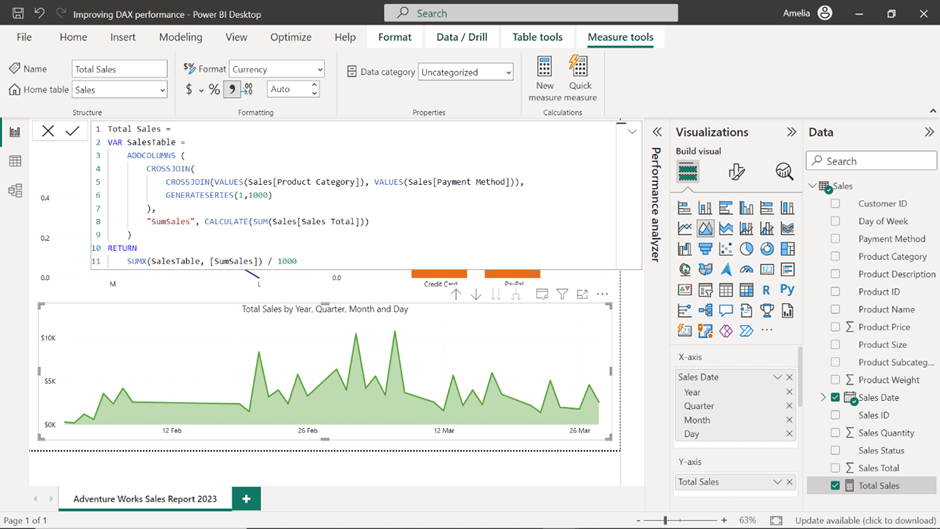


3. In this case, it appears that the **Total Sales by Year, Quarter, Month and Day** area chart is slowing down the report performance, as it has a considerably larger DAX loading time compared to other visuals.

**Step 6: Improve DAX performance**

Now that you have identified that the problematic visual is **Total Sales by Year, Quarter, Month and Day**, the next step is to refine and optimize. This might involve rewriting certain parts of the DAX formula for efficiency, eliminating unnecessary calculations, or simplifying complex ones. The goal is to reduce the computational load on the Power BI engine.

1. Locate the **Total Sales** field from the **Table** view on your right and select it to view the underlying DAX formula. This DAX formula inflates data with the nested **CROSSJOIN** operations, creating a much larger table. For each row, it recalculates the **Total Sales** using **CALCULATE**, a resource-intensive operation. Aggregating this massive table again with **SUMX** further strains performance, making the entire computation slow.



2. To simplify the DAX formula, eliminate the nested **CROSSJOIN** and **GENERATESERIES** functions. Instead use the **SUMX** function enclosed with a single **CROSSJOIN**:

1

2

3

4

5

6

Total Sales =

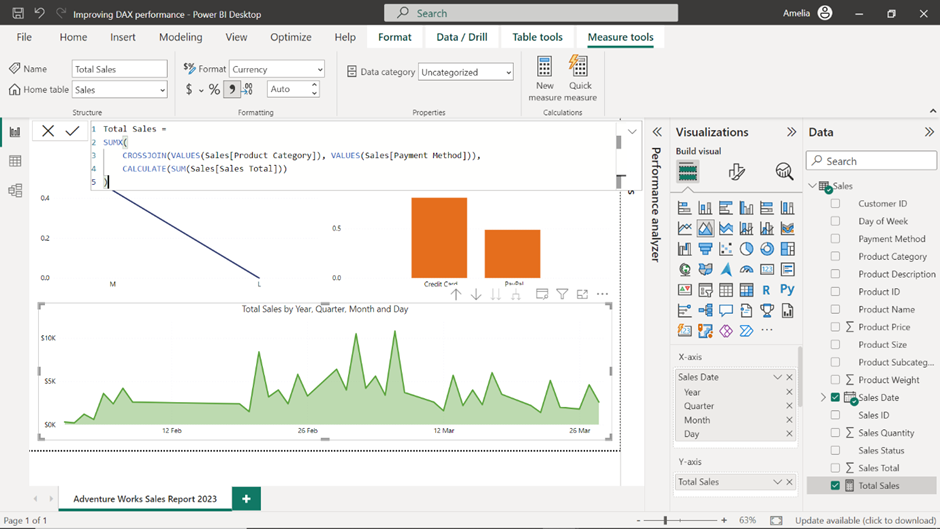
SUMX(

    CROSSJOIN(VALUES(Sales[Product Category]), VALUES(Sales[Payment Method])),

    CALCULATE(SUM(Sales[Sales Total]))

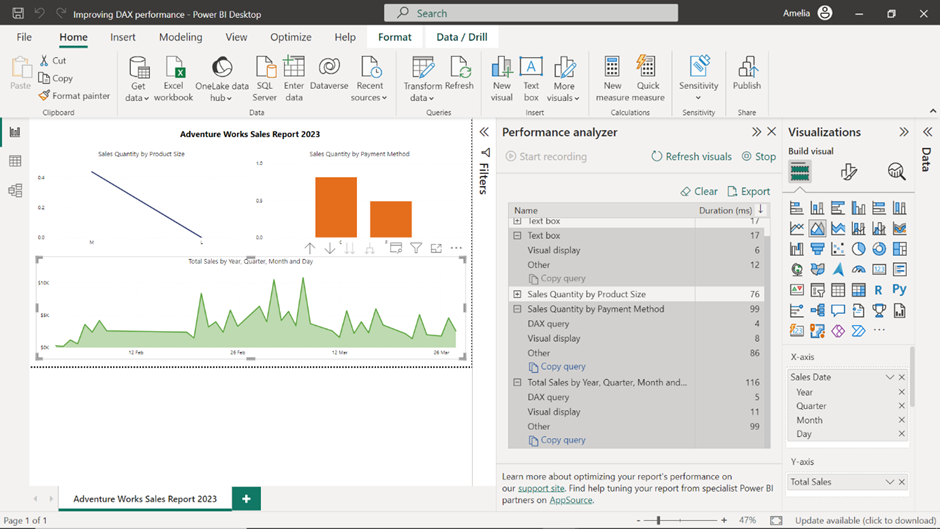
)

3. The **SUMX** function iterates over each unique combination of **Product Category** and **Payment Method** and calculates the **Total Sales** for that specific combination using **CALCULATE (SUM (Sales [Sales Total]))**. After calculating the sales for each combination, **SUMX** then sums up these individual results to produce the final **Total Sales**.



**Step 7: Test and review**

* Finally, re-run the **Performance Analyzer** to test if the optimization was successful. Optimizing DAX is often an iterative process. Once changes are made, it's vital to test and review the impact. Sometimes, what seems like an optimization might not have the desired effect, or there could be unintended side effects. Regular testing ensures that you're moving in the right direction.



**Conclusion**

After walking through these steps to enhance your DAX performance, you've not only elevated the efficiency of your report but acquired a set of skills that are universally applicable. Always remember, in the world of data analytics, it's not just about having data, but how swiftly and accurately you can harness its potential.