Big Problem, Small Solution: Using USERELATIONSHIP to Solve Performance Issues in Power BI

In the world of data analysis, especially in complex industries like distribution, building efficient dashboards can be a daunting task. I was recently asked to create a dashboard that tracks the performance hygiene of multiple distribution businesses. We called this dashboard **"Rise of Discipline."**

The Challenge: Monitoring Performance Hygiene

The key performance indicators (KPIs) we needed to track were quite comprehensive, including:

1. Daily on-time presence of all employees at each distribution house.
2. Daily Sales Target vs. Achievement (TGT vs ACH).
3. Daily on-time (within Service Level Agreement - SLA) sales confirmation.
4. Daily on-time deposit confirmation.
5. Weekly petty expense entry (at least two times per week).
6. Monthly budget vs. expense status.

To complicate things further, we could only measure performance on **weekdays**. Holidays and weekends were excluded, but we still needed to track monthly targets on a month-to-date (MTD) basis, regardless of these exclusions.

Tailoring KPIs to Business Needs

For each KPI, we set up customized calculations. For example, the **SLA times** vary across different distribution houses depending on the business sector. Telecommunication houses, for example, close daily operations before the evening, while FMCG (fast-moving consumer goods) businesses might take longer. I had to ensure that the SLA for each KPI was tailored specifically to the requirements of each distribution house.

The real headache came when it was time to implement the weekly petty cash entry KPI. Unlike other KPIs, this one had specific conditions based on business type, and it created an unexpected **cardinality** issue.

The Core Problem: Handling Multiple Relationships

One challenge involved calculating the number of petty expense entries **based on the expense creation date**. To do this, I needed to link **one dimension table** (the Date table) to **two different date columns** in the same fact table (Incur Date and Expense Creation Date).

However, Power BI only allows **one active relationship** between a dimension and fact table at any given time. So, I had to find a way to connect both date columns without causing conflicts.

My Initial Attempt: Duplicating Tables

At first, I tried duplicating the fact table and establishing a different relationship between the new fact table and the date column in the dimension table. I created a duplicate of the fact table, named it **Fact Copied**, and connected it to the Calendar table through the **Expense Creation Date**.

While this worked, it introduced **performance issues**. Each time I refreshed the data, Power BI had to reload the extra table, which increased storage, refresh times, and added unnecessary complexity to the data model. It was not an ideal solution, and I knew there had to be a better way.

The Small Solution: USERELATIONSHIP to the Rescue

Enter the **USERELATIONSHIP** function in DAX. This function allows you to create **temporary relationships** between two tables **within a measure**. Instead of duplicating tables or creating complex workarounds, I used `USERELATIONSHIP` to activate an inactive relationship between the date and expense creation date columns only when I needed it for my calculations.

Here's how I used it:

```
TotalWeeklyExpenseEntries =
CALCULATE(
    COUNT(FactTable[ExpenseID]),
    USERELATIONSHIP(CalendarTable[Date],
FactTable[ExpenseCreationDate])
)
```

With `USERELATIONSHIP`, I was able to:

- **Activate the inactive relationship** between the Date table and Expense Creation Date.
- Keep the original **Incur Date** relationship intact for all other KPIs.
- Avoid duplicating fact tables or introducing extra storage and refresh issues.

Result: Smooth Performance and No Duplication

The **USERELATIONSHIP** function solved my cardinality problem efficiently. It allowed me to switch between relationships in my measures without adding unnecessary tables or complexity to the model. This not only improved the dashboard's **performance** but also streamlined the process, making the data model easier to manage.

By using `USERELATIONSHIP`, I avoided a **big problem** with a **small solution**, ensuring my distribution businesses' performance hygiene dashboard was both accurate and highly efficient.

In conclusion, the `USERELATIONSHIP` function is an incredibly handy tool in Power BI, especially when you need to manage multiple relationships within the same data model. If you ever find yourself in a similar situation where duplicating tables is causing performance issues, consider this simple yet powerful DAX function to keep your data model lean and efficient!