Great questions! Let's break this down **step by step** in a simple and clear way:

**🎯 What is SUMX in Power BI?**

**SUMX is a DAX function that calculates the sum of an expression, row by row, over a table.**

Think of it as:

“Go to each row in the table, calculate something for that row, then add it all up.”

**🧸 Simple Example:**

Imagine a **Sales Table** like this:

| **Product** | **Quantity** | **Price** |
| --- | --- | --- |
| Apple | 10 | 5 |
| Banana | 20 | 3 |

If you want to calculate **total revenue**, you write:

Total Revenue = SUMX(Sales, Sales[Quantity] \* Sales[Price])

This will do:

* Row 1: 10 \* 5 = 50
* Row 2: 20 \* 3 = 60
* Then adds: **50 + 60 = 110**

✅ So SUMX gives you control over how to calculate and then sum.

**🔍 What is Context in Power BI?**

Power BI has **two types of context**:

**1. Row Context**

Think of this as:

“I’m looking at **one row at a time**.”

This is what happens **inside calculated columns** or when looping through rows using functions like SUMX.

**2. Filter Context**

This is:

“What filters are applied to the report/table/visual right now?”

For example:

* If you're looking at "Sales for Nepal," the filter context is **Country = Nepal**.
* If you add a slicer or click a chart, you're changing the filter context.

**🤝 What is Context Transition?**

This is **the magic bridge between row context and filter context**.

**🧠 Simple Definition:**

When you are inside a row (row context), and you use a **measure or CALCULATE**, Power BI **converts the row context into a filter context**.

**📦 Example of Context Transition:**

Imagine this table:

| **Customer** | **Total Sales** |
| --- | --- |
| John | ? |
| Mary | ? |

You write a measure like:

Total Sales = SUM(Sales[Amount])

If you use CALCULATE([Total Sales]) **inside a row context** (e.g., inside SUMX or a calculated column), Power BI will **convert that row into a filter** like:

* "Hey, filter the Sales table where Customer = John"
* Then calculate the sum of Sales for that customer.

**🔄 How SUMX uses Context Transition**

Let’s say you write:

Total Revenue by Customer =

SUMX(Customers, [Total Sales])

* Customers table has a **row context** (each customer one at a time).
* [Total Sales] is a **measure**, so Power BI **converts the current row’s customer into a filter** (context transition).
* Then it evaluates [Total Sales] for each customer.
* Finally, it **adds all of them up**.

✅ So SUMX **+ a measure** causes **context transition** to happen.

**🧪 Real-Life Use Case:**

**Goal: Calculate total sales after discount from a transactions table**

| **Product** | **Quantity** | **Price** | **Discount** |
| --- | --- | --- | --- |
| A | 2 | 100 | 10 |
| B | 3 | 200 | 20 |

You can write:

Total Net Sales =

SUMX(

Sales,

(Sales[Quantity] \* Sales[Price]) - Sales[Discount]

)

* It calculates per row: (2×100 -10), (3×200 - 20)
* Then adds them all.

**🔚 Summary:**

| **Concept** | **Simple Meaning** |
| --- | --- |
| SUMX | Go row by row, calculate something, then sum |
| Row Context | You're inside a table row |
| Filter Context | Filters applied from visuals/slicers |
| Context Transition | Converts row context into filter context (usually when using a measure inside SUMX or CALCULATE) |

Let me know the **problem you're solving** and I can help you write the exact SUMX formula for it!

Great! You're exploring two different versions of SUMX — both **do similar work** but **operate in different contexts**. Let's break down what you're doing and **when to use each** version in a kid-friendly and clear way.

**🎯 1. Version 1:**

Total Sales = SUMX(Purchase, Purchase[Price] \* Purchase[Quantity])

**✅ What it does:**

* Goes **row by row** in the **Purchase table**.
* Multiplies Price × Quantity for each row.
* Sums it all up.

**🧠 When to use:**

* When you're calculating **total sales across all purchases** — not filtered by any other table.

It’s like saying: “Let me go through the full purchase log and total everything up.”

**🎯 2. Version 2:**

Total Sales = SUMX(RELATEDTABLE(Purchase), Purchase[Price] \* Purchase[Quantity])

**✅ What it does:**

* RELATEDTABLE(Purchase) gets **only the rows in Purchase table that are related to the current row** (like Customer, Product, etc.).
* Then SUMX works on **just those related rows**.

**🧠 When to use:**

* When you're writing this measure **inside another table context** (like Customer, Product, Salesperson).
* For example, if you're in a Customer table and want to know total sales per customer.

It’s like saying: “I’m looking at John (a customer). Show me **only John's purchases**, and then total those.”

**📊 Use Case Example:**

**You have these tables:**

1. **Customer Table**  
   | CustomerID | CustomerName |  
   |------------|--------------|  
   | 1 | Alice |  
   | 2 | Bob |
2. **Purchase Table**  
   | PurchaseID | CustomerID | Price | Quantity |  
   |------------|------------|-------|----------|  
   | 101 | 1 | 100 | 2 |  
   | 102 | 1 | 50 | 1 |  
   | 103 | 2 | 200 | 1 |

**✅ Measure using RELATEDTABLE():**

In the **Customer Table**, you write:

Total Sales = SUMX(RELATEDTABLE(Purchase), Purchase[Price] \* Purchase[Quantity])

This will:

* For Alice: sum (100×2) + (50×1) = **250**
* For Bob: sum (200×1) = **200**

So it gives you **sales by customer**, thanks to context transition and filtering with RELATEDTABLE.

**🔍 Summary Table:**

| **Formula** | **Used In** | **Works On** | **Filters Based On** |
| --- | --- | --- | --- |
| SUMX(Purchase, ...) | Any context | All rows in Purchase | None (unless slicer applied) |
| SUMX(RELATEDTABLE(Purchase), ...) | When inside a related table (e.g. Customer) | Only related rows | Automatically filtered by row context |

Let me know what you're trying to calculate specifically (e.g. by customer, product, etc.), and I can give you the exact version that fits best!