**Sales Data Analysis  
Power BI Project Documentation**

Dataset + Data Model + Measures + Report Design

Last updated: January 08, 2026

Made by: Mgr. Tamás Csiba

# Table of Contents

1. Project Overview

2. Dataset Overview

3. Data Model

4. Data Dictionary

5. Data Preparation (Power Query)

6. Data Model (Relationships)

7. Report Page – Top/Bottom 5 Products (Sales, Quantity, Profit)

8. Report Page – Sales Trends Over Time (Day / Month / Quarter / Year)

9. Report Page – Relationship Between Profit and Net Sales

# 1. Project Overview

This Power BI project provides an interactive dashboard for analyzing sales performance. It supports high-level monitoring (KPIs, trends, top/bottom products) and detailed drill-down to the order level.

## Business Questions Covered

* Top/Bottom 5 products by Sales, Profit, and Quantity Sold
* Sales trends over time (daily, monthly, quarterly, annually)
* Relationship between Sales and Profit
* Comparison of Sales/Profit/Quantity between two user-selected periods
* Average discount per discount category
* Total number of orders
* Order-level table with full details and visual filters (Product, Date, Customer, Promotion, etc.)
* Sales by city

# 2. Dataset Overview

Source file: Store\_Data.xlsx

|  |  |
| --- | --- |
| Field | Value |
| Workbook | Store\_Data.xlsx |
| Fact table | Sheet3 |
| Fact rows | 3510 |
| Date range | 2020-01-01 to 2024-01-01 |
| Customers | 50 |
| Products | 30 |
| Promotions | 5 |
| Distinct customers in fact | 50 |
| Distinct products in fact | 30 |
| Promotion usage | 0 = No Promotion (most rows), PR001-PR005 = active promotions |

The workbook contains three dimension tables (Customers, Product, Promotion) and one fact table (Sheet3). Sheet3 is the transactional dataset and is the base for building measures and visuals.

## Grain

One row in the fact table represents a single sales transaction line with a Date, Customer, Product, Promotion (or none), and Units Sold.

# 3. Data Model

Star schema model:

* Fact Sales (from Sheet3)
* Dim Customers (from Dim Customers)
* Dim Product (from Dim Product)
* Dim Promotion (from Dim Promotion) + a synthetic 'No Promotion' row (PromotionID = 0)

## Relationships

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| From (Table) | Column | To (Table) | Column | Cardinality |
| Fact Sales | CustomerID | Dim Customers | Customer ID | Many-to-one |
| Fact Sales | Product ID | Dim Product | ProductID | Many-to-one |
| Fact Sales | PromotionID | Dim Promotion | PromotionID | Many-to-one (0 = No Promotion) |
| Fact Sales | Date | Dim Date | Date | Many-to-one |

# 4. Data Dictionary

## Dim Customers

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Type | Example | Notes |
| Customer ID | Whole number | 1 | Primary key. Join to Fact Sales[CustomerID]. |
| Customer Name | Text | Aarav Singh | Customer full name |
| City | Text | Nagpur | City name. Values contain trailing spaces in the raw file |
| State | Text | Maharashtra | State/region. Values contain trailing spaces in the raw file |
| Pincode | Whole Number | 440001 | Postal code. Keep as text if you want to preserve formatting. |
| EmailID | Text | 1Aarav@gmail.com | Customer email. |
| Phone Number | Whole number | 860135097 | Phone number. |

## Dim Product

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Type | Example | Notes |
| ProductID | Text | P001 | Primary key. Join to Fact Sales[Product ID]. |
| Product Name | Text | Apple iPhone 14 | Product display name. |
| Product Line | Text | Electronics | High-level category (e.g., Electronics). |
| Price (INR) | Whole number | 79999 | Unit price in Indian Rupees (INR). |

## Dim Promotion

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Type | Example | Notes |
| PromotionID | Text | PR001 | Primary key. Join to Fact Sales[PromotionID]. Add a 'No Promotion' row for ID 0. |
| Promotion Name | Text | Summer Sale | Promotion name. |
| Ad Type | Text | Email | Channel (Email, Social Media, etc.). |
| Coupon Code | Text | SUMMER21 | Coupon code used by promotion. |
| Price Reduction Type | Whole number | 20% off | Discount rule (e.g., '20% off' or 'Buy 1 Get 1 Free'). |

## Fact Table (Sheet3)

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Type (recommended) | Example | Notes |
| Date (dd/mm/yyyy) | Date | 2020-02-04 | Order date. |
| CustomerID | Text | 10 | Foreign key to Dim Customers[Customer ID]. |
| PromotionID | Text | 0 | Foreign key to Dim Promotion[PromotionID]. '0' means no promotion. |
| Product ID | Text | P004 | Foreign key to Dim Product[ProductID]. |
| Units Sold | Whole number | 2 | Quantity sold (1-3 in this dataset). |
| Price Per Unit | Whole number |  | Blank in source - derive from Dim Product[Price (INR)]. |
| Total Sales | Whole number |  | Blank in source - derive as Units Sold \* Price Per Unit. |
| Discount Percentage | Whole number |  | Blank in source - derive from promotion rule (percent/BOGO). |
| Discount Value | Whole number |  | Blank in source - derive as Total Sales \* Discount %. (BOGO requires rule.) |
| Net Sales | Whole number |  | Blank in source - derive as Total Sales - Discount Value. |

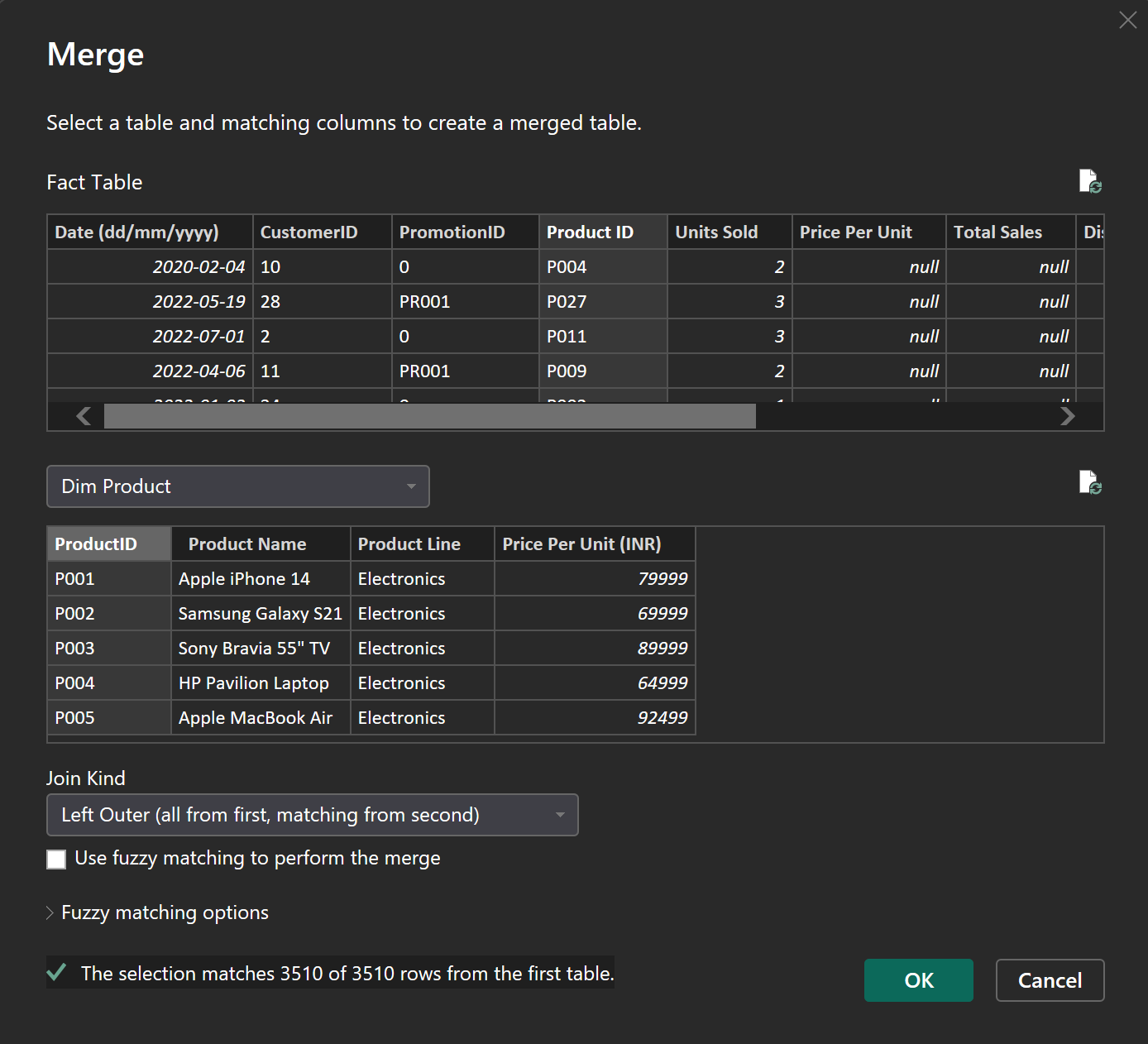
# 5. Data Preparation (Power Query)

# Power Query – Fixing Null Values (Derived Columns)

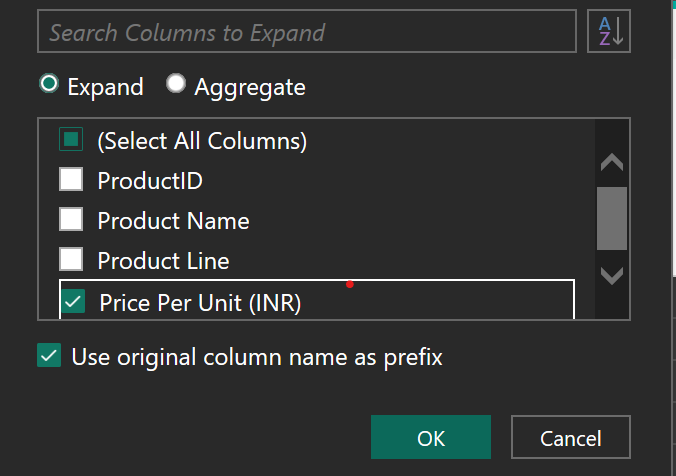
In the original Excel dataset, some measure-like fields (e.g., Total Sales, Discount Percentage/Value, Net Sales) were missing or contained nulls. To ensure consistent reporting and accurate calculations, these values were recreated in Power Query using merges to dimension tables and custom columns.

## 1) Restore Price Per Unit via Merge (Dim Product)

Because the fact table contained null values for Price Per Unit, we merged the Fact table with Dim Product on ProductID and expanded the Price Per Unit (INR) field.



*Figure 1 – Merge Fact Table with Dim Product to bring Price Per Unit (INR).*



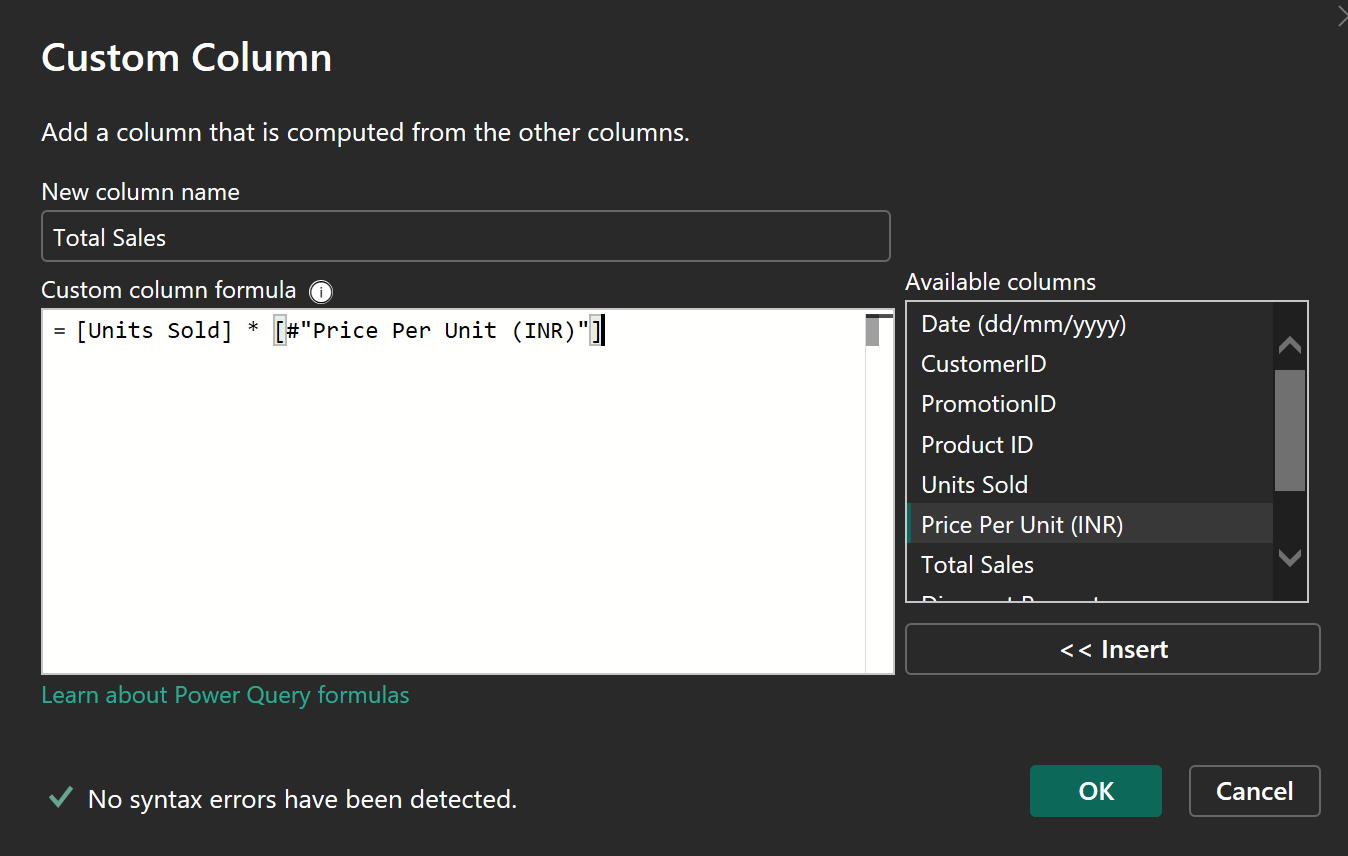
*Figure 2 – Expand only the required column (Price Per Unit (INR)).*

## 2) Create Total Sales (Units Sold × Price Per Unit)

After restoring Price Per Unit, Total Sales can be computed directly in Power Query as Units Sold × Price Per Unit.

Custom Column formula (Power Query M):

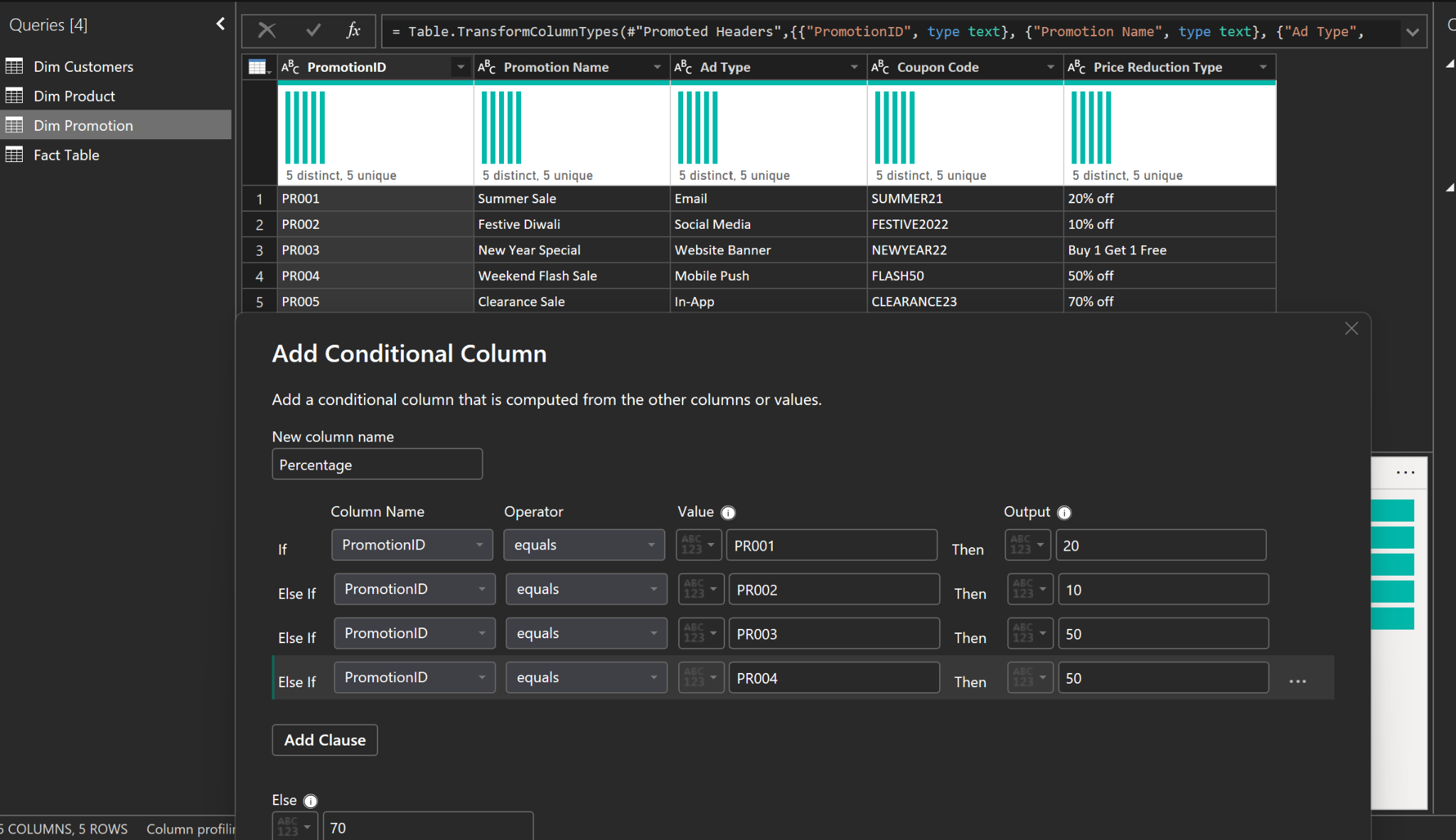
= [Units Sold] \* [#"Price Per Unit (INR)"]



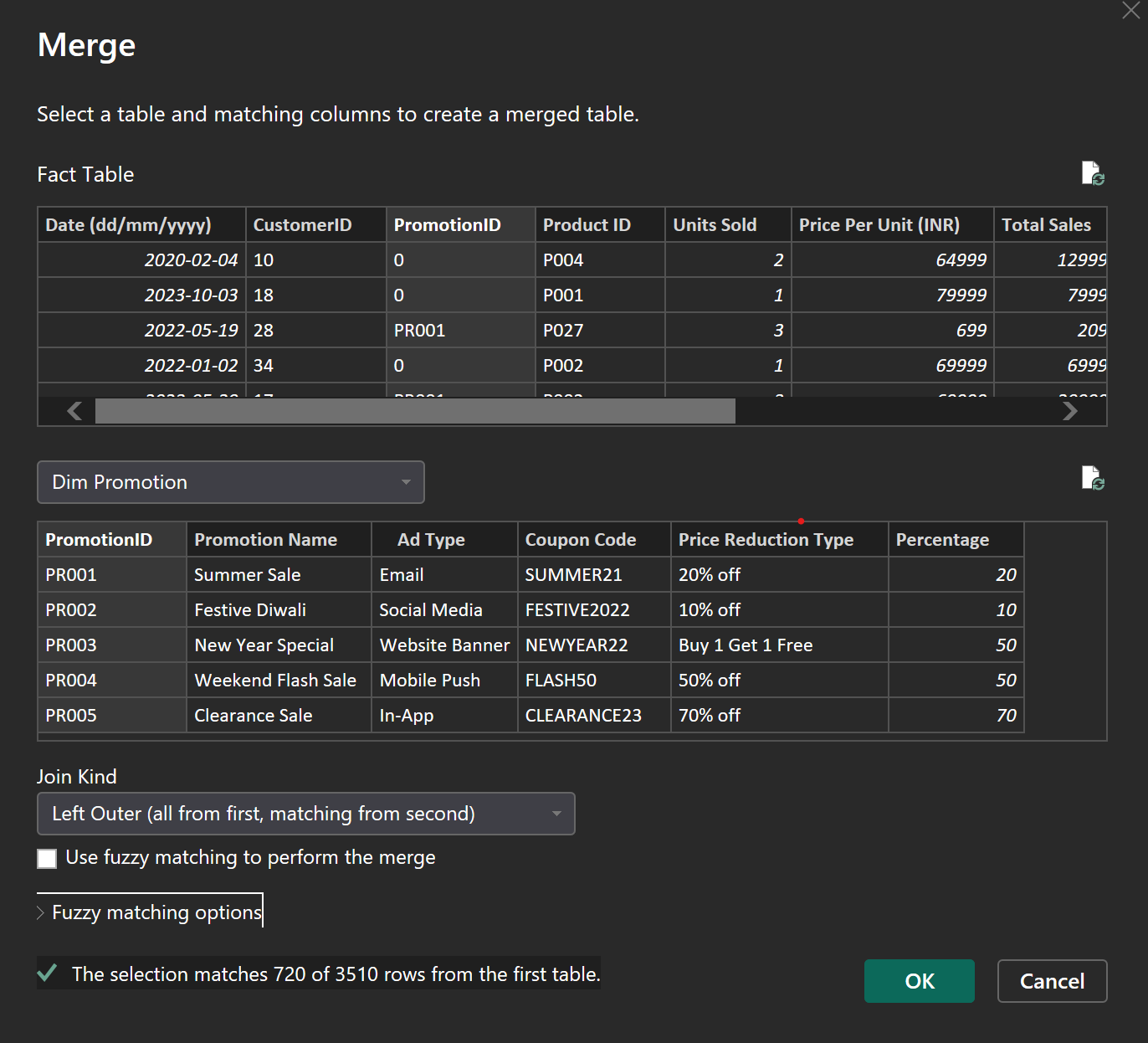
*Figure 3 – Create Total Sales as Units Sold × Price Per Unit.*

## 3) Restore Discount Percentage via Merge (Dim Promotion)

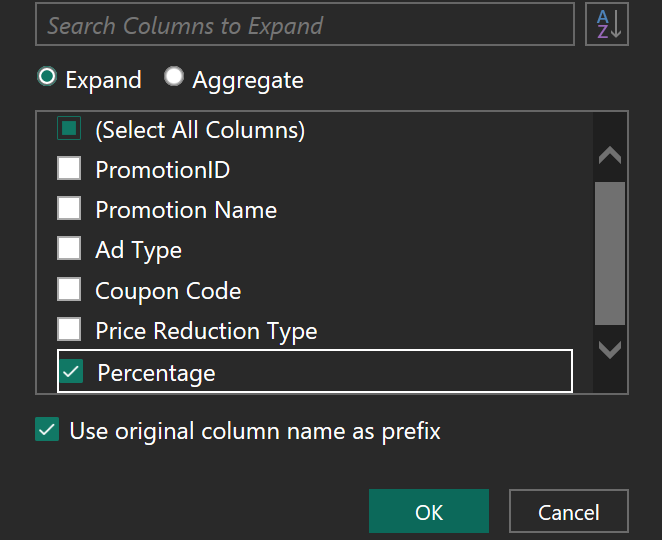
Discount Percentage was reconstructed based on PromotionID. In Dim Promotion, a conditional column was created to map PromotionID to a numeric percentage (e.g., PR001 → 20, PR002 → 10, etc.). Then the Fact table was merged with Dim Promotion on PromotionID and the Percentage column was expanded.



*Figure 4 – Create Percentage in Dim Promotion using a conditional column.*



*Figure 5 – Merge Fact Table with Dim Promotion on PromotionID (Left Outer join).*



*Figure 6 – Expand only the Percentage column into the Fact table.*

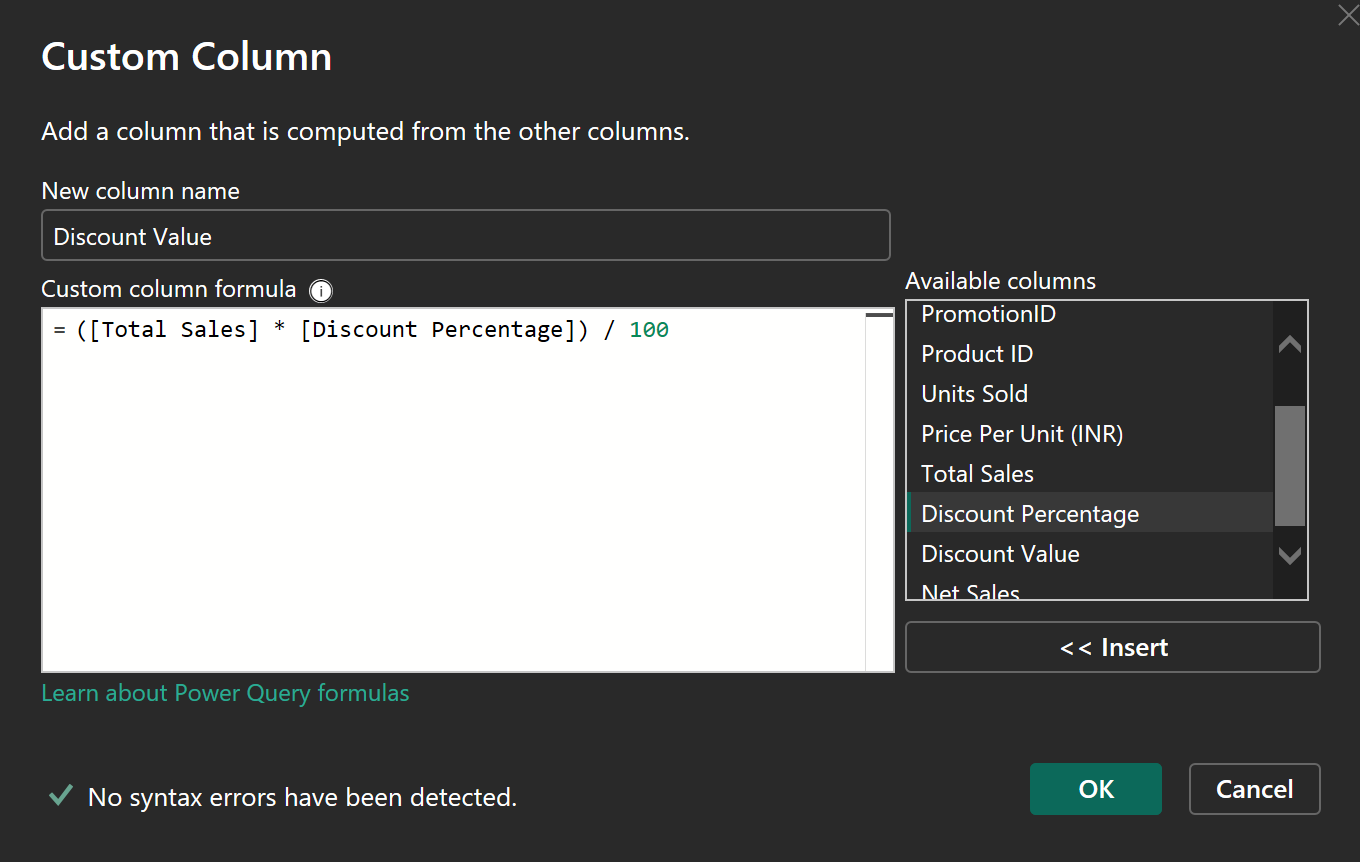
## 4) Create Discount Value (Total Sales × Discount Percentage / 100)

Once Total Sales and Discount Percentage are available in the fact table, Discount Value is calculated as:

Discount Value = (Total Sales × Discount Percentage) / 100

Custom Column formula (Power Query M):

= ([Total Sales] \* [Discount Percentage]) / 100



*Figure 7 – Create Discount Value as (Total Sales × Discount Percentage) / 100.*

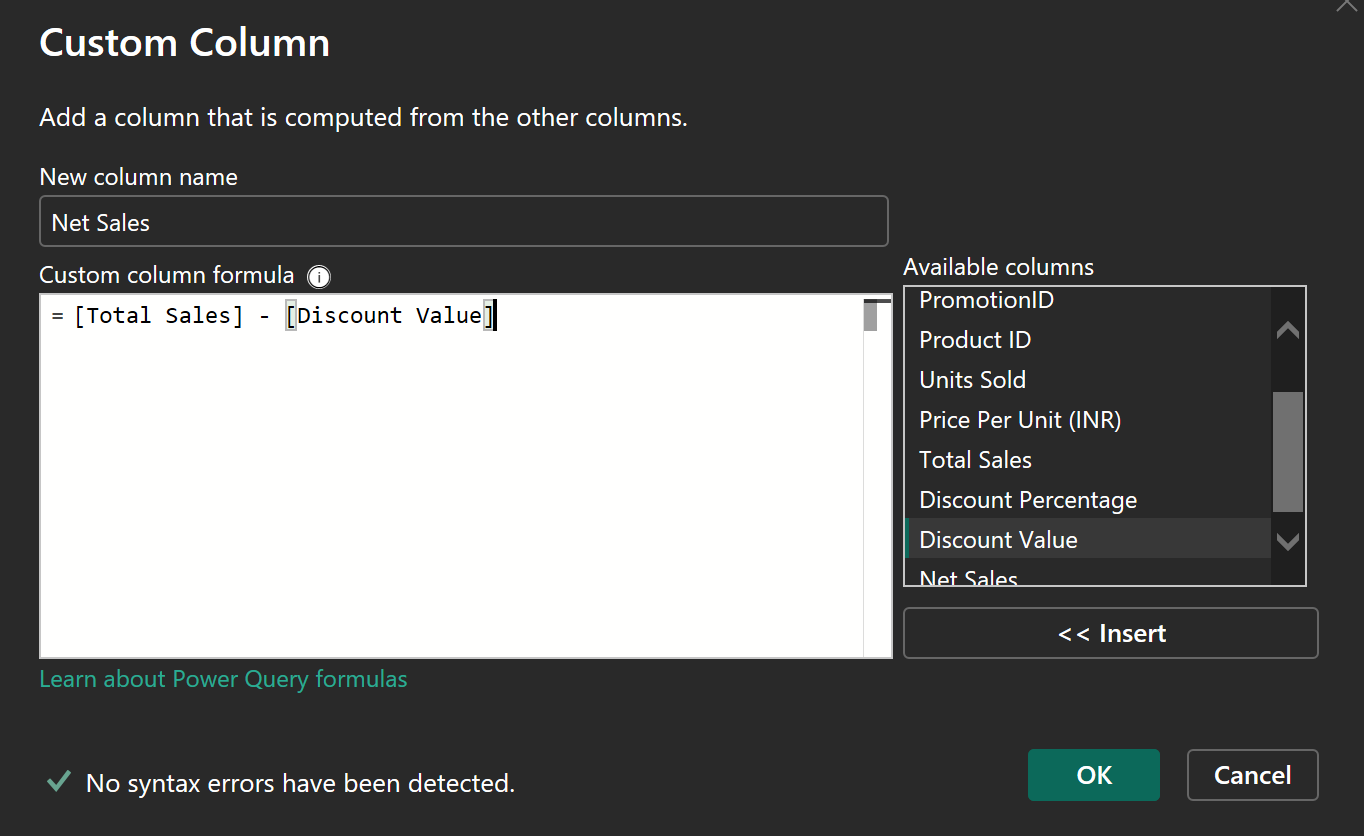
## 5) Create Net Sales (Total Sales − Discount Value)

Net Sales is derived by subtracting Discount Value from Total Sales.

Net Sales = Total Sales − Discount Value

Custom Column formula (Power Query M):

= [Total Sales] - [Discount Value]



*Figure 8 – Create Net Sales as Total Sales − Discount Value.*

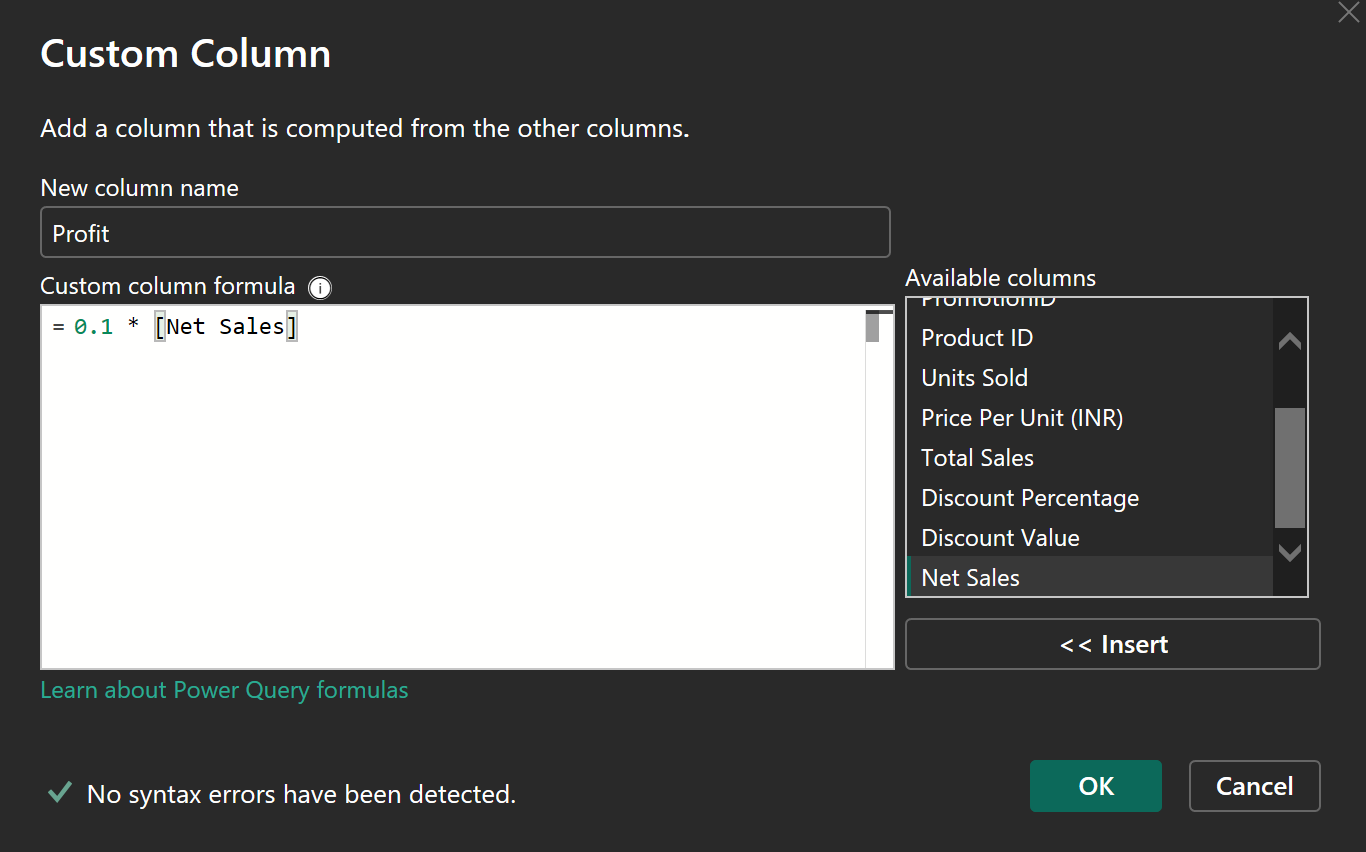
## 6) Create Profit (Assumed 10% of Net Sales)

For this project, Profit is calculated as a simple margin on Net Sales. We assume a 10% profit margin, therefore:

Profit = 0.1 × Net Sales

Custom Column formula (Power Query M):

= 0.1 \* [Net Sales]



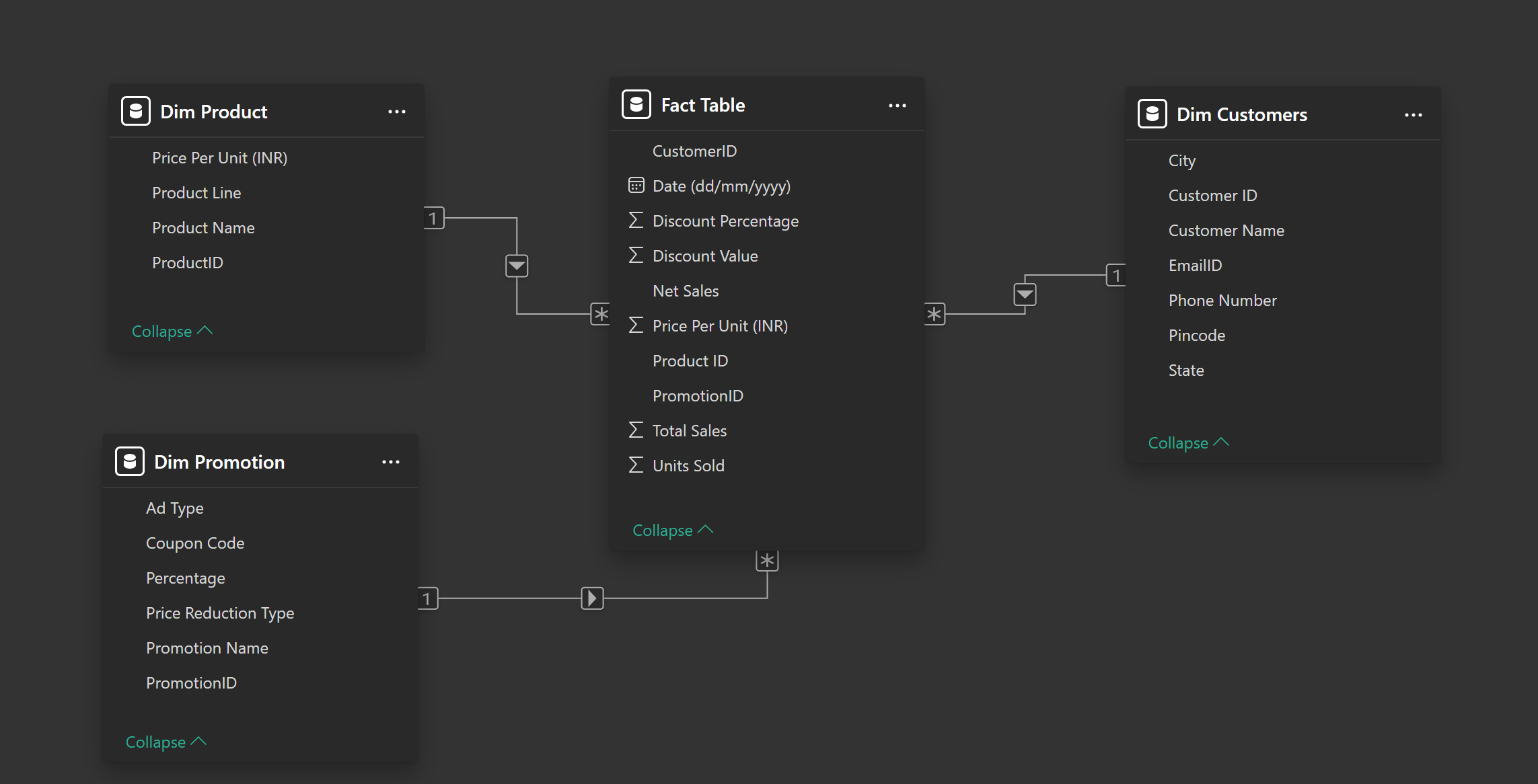
*Figure 9 – Create Profit as 10% of Net Sales (custom column in Power Query).*

# 6. Data Model (Relationships)

The Power BI report is built on a **star schema** consisting of one central **Fact Table** and three supporting dimension tables: **Dim Product**, **Dim Customers**, and **Dim Promotion**.  
This model enables filtering and slicing sales metrics by product attributes, customer location, and promotion details while keeping measures consistent across visuals.

**Relationship Details**

| **Dimension Table** | **Dimension Key** | **Fact Table Key** | **Cardinality** | **Cross-filter direction** |
| --- | --- | --- | --- | --- |
| Dim Product | ProductID | Product ID | 1 → \* | Single (Dim → Fact) |
| Dim Customers | Customer ID | CustomerID | 1 → \* | Single (Dim → Fact) |
| Dim Promotion | PromotionID | PromotionID | 1 → \* | Single (Dim → Fact) |

****

*Figure 10 – Data model relationships (Fact Table → Dim Product / Dim Customers / Dim Promotion).*

**Notes**

* The **Fact Table** stores the transactional grain (date × customer × product × promotion).
* Dimension tables provide descriptive attributes used in slicers and drill-downs:
  + **Dim Product:** Product Name, Product Line, Price Per Unit (INR)
  + **Dim Customers:** City, State, Pincode, etc.
  + **Dim Promotion:** Promotion Name, Ad Type, Coupon Code, Percentage
* Using **single-direction filtering** (dimension → fact) prevents ambiguous filter paths and supports reliable aggregations (e.g., Total Sales, Discount Value, Net Sales).

# 7. Top/Bottom 5 Products (Sales, Quantity, Profit)

This page highlights the best- and worst-performing products across three key metrics: Sales, Quantity Sold, and Profit. It is designed for quick product performance benchmarking and to identify outliers that may require pricing, promotion, or inventory actions.

## Visuals on the Page

|  |  |  |  |
| --- | --- | --- | --- |
| Visual | Type | Axis | Value (Measure) |
| Top 5 Products by Sales | Clustered bar chart | Dim Product[Product Name] | SUM(Fact Table[Total Sales]) |
| Bottom 5 Products by Sales | Clustered bar chart | Dim Product[Product Name] | SUM(Fact Table[Total Sales]) |
| Top 5 Products by Quantity | Clustered bar chart | Dim Product[Product Name] | SUM(Fact Table[Units Sold]) |
| Bottom 5 Products by Quantity | Clustered bar chart | Dim Product[Product Name] | SUM(Fact Table[Units Sold]) |
| Top 5 Products by Profit | Clustered bar chart | Dim Product[Product Name] | SUM(Fact Table[Profit]) |
| Bottom 5 Products by Profit | Clustered bar chart | Dim Product[Product Name] | SUM(Fact Table[Profit]) |

Note: In the screenshot, the bottom-right visual title appears as “Top 5 Products by Profit”. For clarity, it should be renamed to “Bottom 5 Products by Profit”.

## Top N / Bottom N Logic

Each visual uses a \*\*visual-level filter\*\* on Product Name with a \*\*Top N\*\* filter:

**• Top charts:** Top N = 5, by the corresponding measure (Sales / Units Sold / Profit).

**• Bottom charts:** Bottom N = 5 (implemented either via Bottom N filter or by sorting ascending and applying Top N = 5), by the corresponding measure.

The charts are sorted so the highest (or lowest) product is clearly visible at the top of each list. Data labels are enabled to display the aggregated value per product.

## Example Output (from current dataset)

In the current dataset snapshot, premium electronics dominate the Top 5 by Sales and Top 5 by Profit (since Profit is proportional to Net Sales). The Bottom 5 lists contain low-revenue items with much smaller totals. Quantity rankings can differ from Sales due to unit price differences.

## How the Top/Bottom 5 Filter Was Implemented (Visual-Level Filter)

Top/Bottom lists were created using a \*\*visual-level Top N filter\*\* on the Product Name field (Filters pane → Filters on this visual). The configuration follows the steps below.

Select the chart (visual) you want to filter.

In the \*\*Filters\*\* pane, under \*\*Filters on this visual\*\*, add/select \*\*Product Name\*\*.

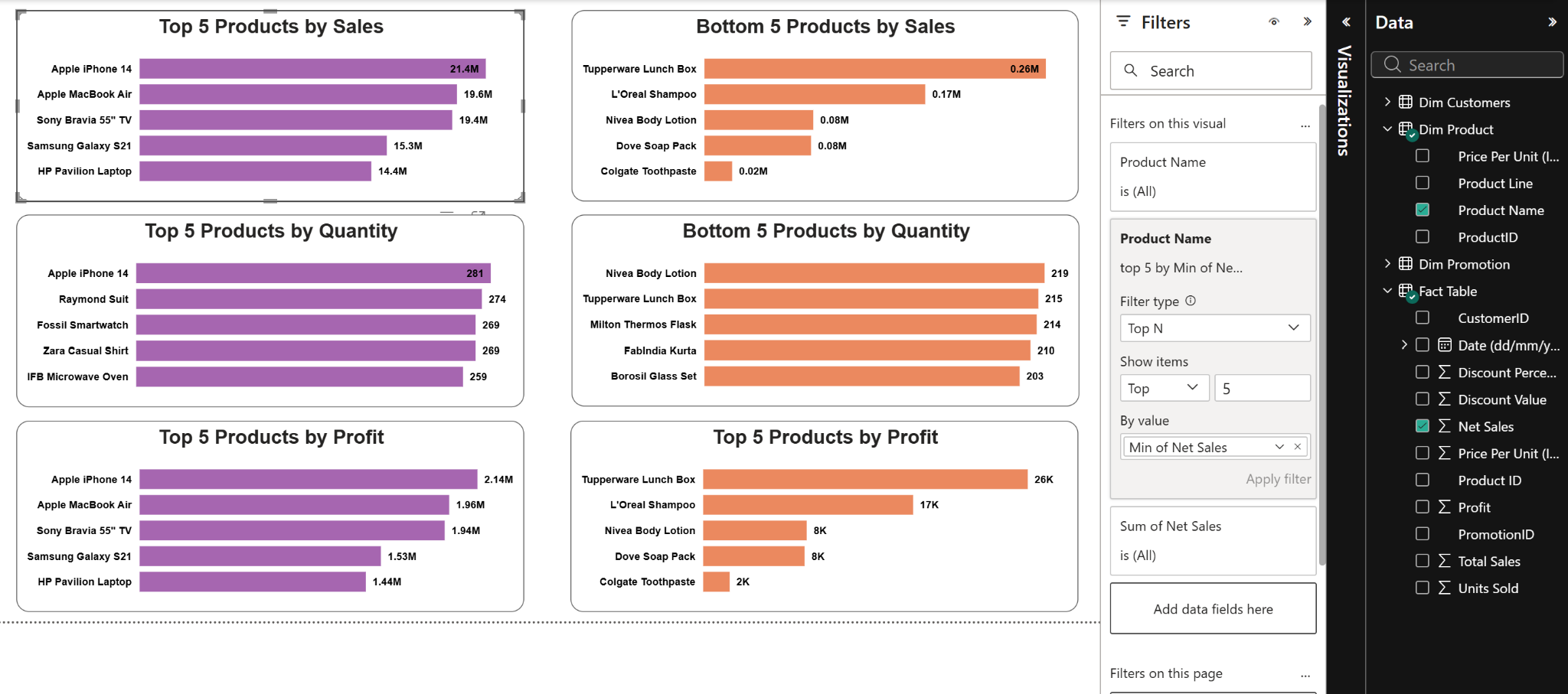
Set \*\*Filter type\*\* to \*\*Top N\*\*.

Set \*\*Show items\*\* to \*\*Top 5\*\* (or \*\*Bottom 5\*\* for bottom charts).

In \*\*By value\*\*, choose the metric to rank by. In the current implementation shown below, \*\*Min of Net Sales\*\* is used as the ranking value.

Click \*\*Apply filter\*\*.

Note: If your goal is Top/Bottom by \*\*Sales\*\*, \*\*Quantity\*\*, and \*\*Profit\*\*, it is recommended to set \*\*By value\*\* to the corresponding measure (e.g., Sum of Total Sales, Sum of Units Sold, Sum of Profit) and use Bottom 5 for the bottom charts. However, the screenshot below documents the exact filter setup used in this version.



*Figure 12 – Visual-level Top N filter configuration (Product Name → Top N → Top 5 by Min of Net Sales → Apply filter).*

# 8. Sales Trends Over Time (Day / Month / Quarter / Year)

This page provides a time-series view of sales performance at multiple granularities. It enables quick identification of seasonality patterns, peaks/dips, and broader performance changes over time. The visuals are built using the Date field’s hierarchy (Year → Quarter → Month → Day) and a single sales metric to keep comparisons consistent across time levels.

## Visuals on the Page

|  |  |  |  |
| --- | --- | --- | --- |
| Visual Title | Visual Type | X-axis | Y-axis (Measure) |
| Sales Trends by Days | Line chart | Date hierarchy → Day | Sum of Net Sales |
| Sales Trends by Months | Line chart | Date hierarchy → Month | Sum of Net Sales |
| Sales Trends by Quarters | Line chart | Date hierarchy → Quarter | Sum of Net Sales |
| Sales Trends by Years | Line chart | Date hierarchy → Year | Sum of Net Sales |

## Fields and Configuration

All four line charts use the same measure on the Y-axis to make the trend comparable across different time grains.

X-axis configuration:

Field: `Date (dd/mm/yyyy)` from the Fact Table.

Hierarchy levels used: Year, Quarter, Month, Day (depending on the chart).

The built-in date hierarchy is used to represent each granularity without duplicating fields.

Y-axis configuration:

• Measure: \*\*Sum of Net Sales\*\* (aggregated from the Fact Table).

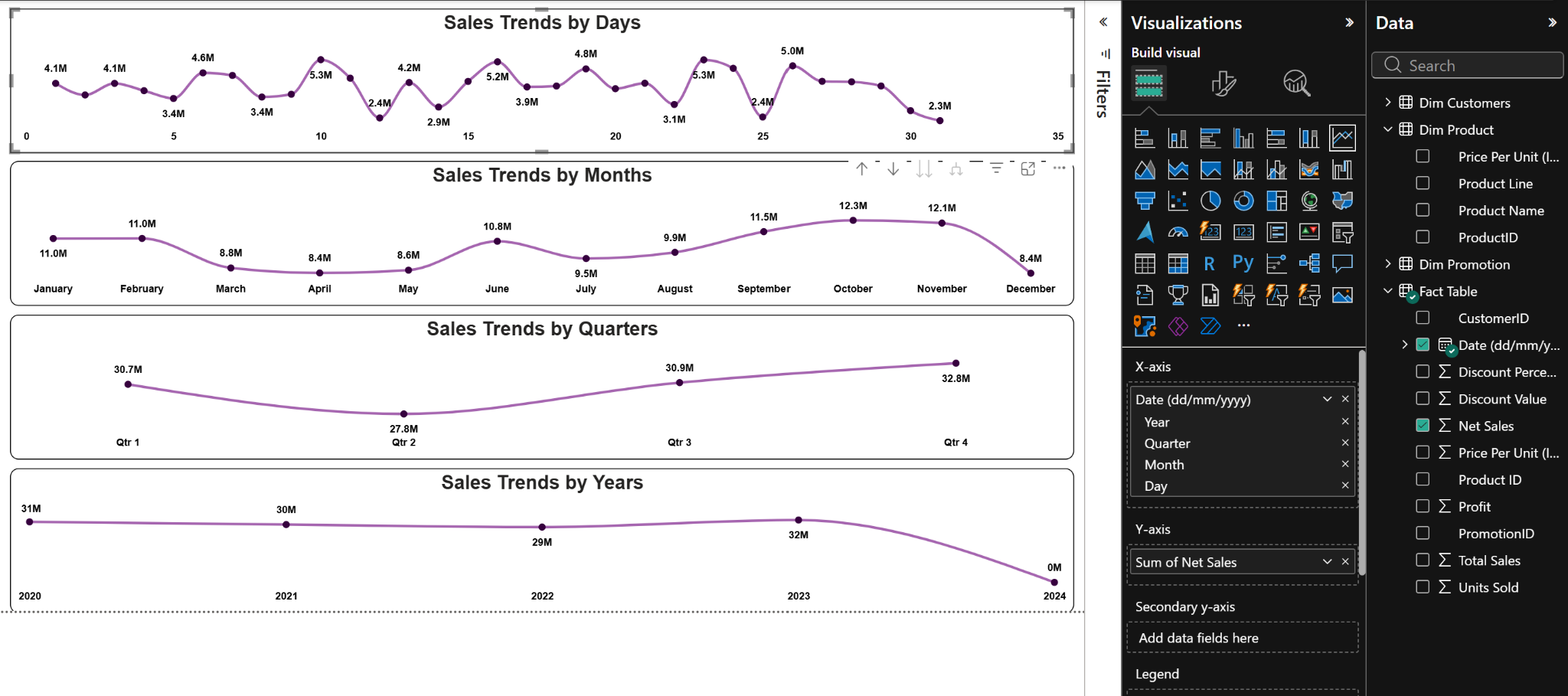
## Interpretation Notes

• Daily trends help identify short-term spikes and irregularities.

• Monthly trends highlight seasonality (e.g., promotional periods, end-of-year demand).

• Quarterly trends provide a stable view for business reporting.

• Yearly trends support long-term performance comparisons.



*Figure 13 – Sales trends page showing Net Sales by Day, Month, Quarter, and Year.*

# 9. Relationship Between Profit and Net Sales

This page visualizes the relationship between \*\*Profit\*\* and \*\*Net Sales\*\* using a scatter plot. The objective is to understand how profit scales with revenue and to identify potential outliers (e.g., unusually low profit for a given net sales value).

## Visual on the Page

|  |  |  |  |
| --- | --- | --- | --- |
| Visual Title | Visual Type | X-axis | Y-axis |
| Profit vs. Net Sales | Scatter chart | Profit | Net Sales |

## Fields and Configuration

Configuration as shown in the report:

• Visual type: Scatter chart

• X-axis: `Profit` (Fact Table)

• Y-axis: `Net Sales` (Fact Table)

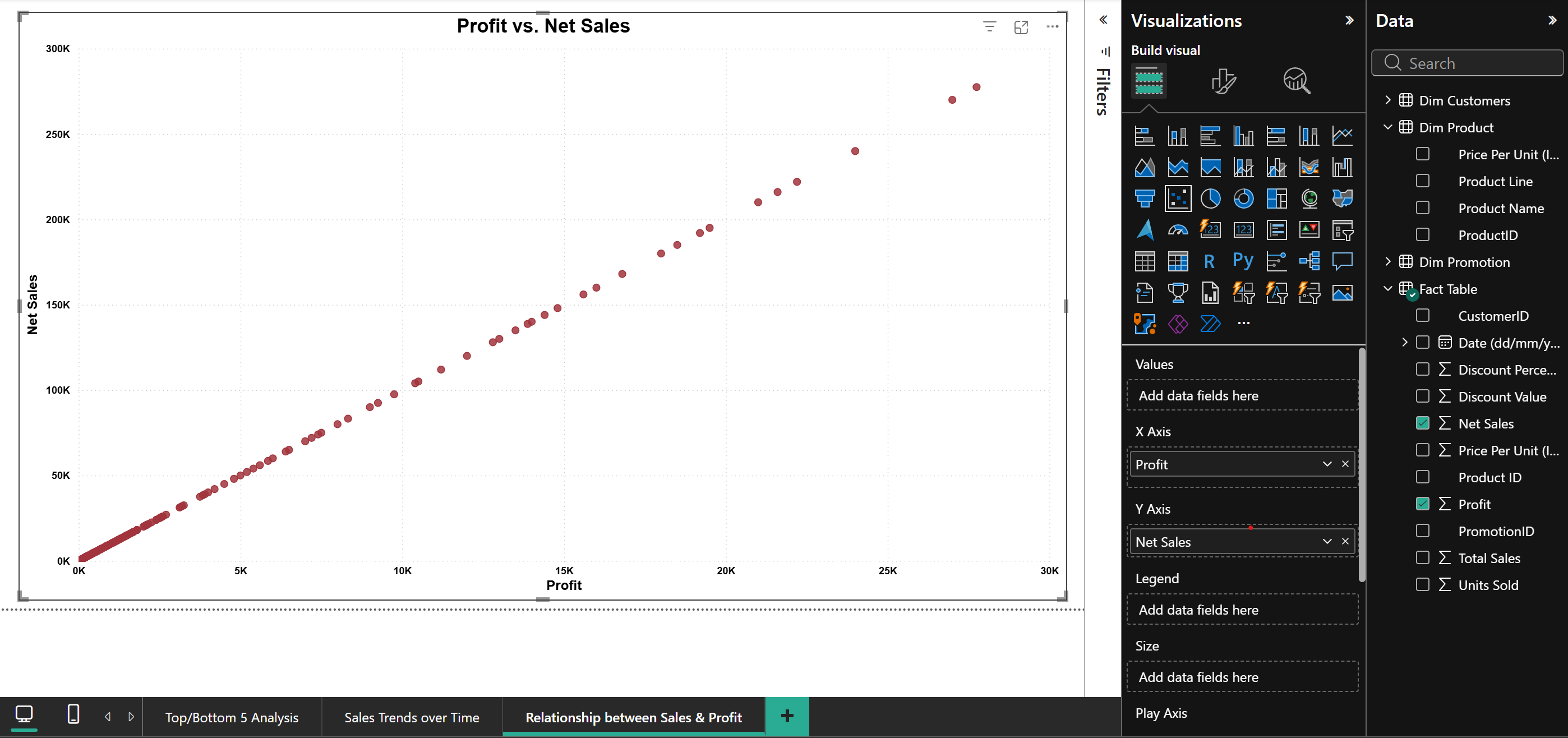
• Legend: not used in the current version

• Size: not used in the current version

## Interpretation Notes

In this dataset, Profit was derived in Power Query as \*\*10% of Net Sales\*\*. Because of this business rule, the scatter points form a near-perfect linear relationship (Profit increases proportionally with Net Sales).

If a more realistic profitability analysis is required, Profit should be calculated from product costs / margins (or a margin table) rather than a fixed percentage of Net Sales. In that case, this scatter plot becomes useful for spotting outliers where Profit deviates from the expected range.



*Figure 13 – Sales trends page showing Net Sales by Day, Month, Quarter, and Year.*

# 10. Average Discount by Promotion Category

This page compares promotional activity by showing the \*\*average discount value\*\* for each promotion category. It provides a quick view of which campaigns typically provide larger discounts and can be used to evaluate promotion strategy.

## Visual on the Page

|  |  |  |  |
| --- | --- | --- | --- |
| Visual Title | Visual Type | X-axis | Y-axis (Aggregation) |
| Average of Discount by Product Categories | Clustered column chart | Dim Promotion[Promotion Name] | Average of Fact Table[Discount Value] |

## Fields and Configuration

• X-axis: `Promotion Name` (Dim Promotion)

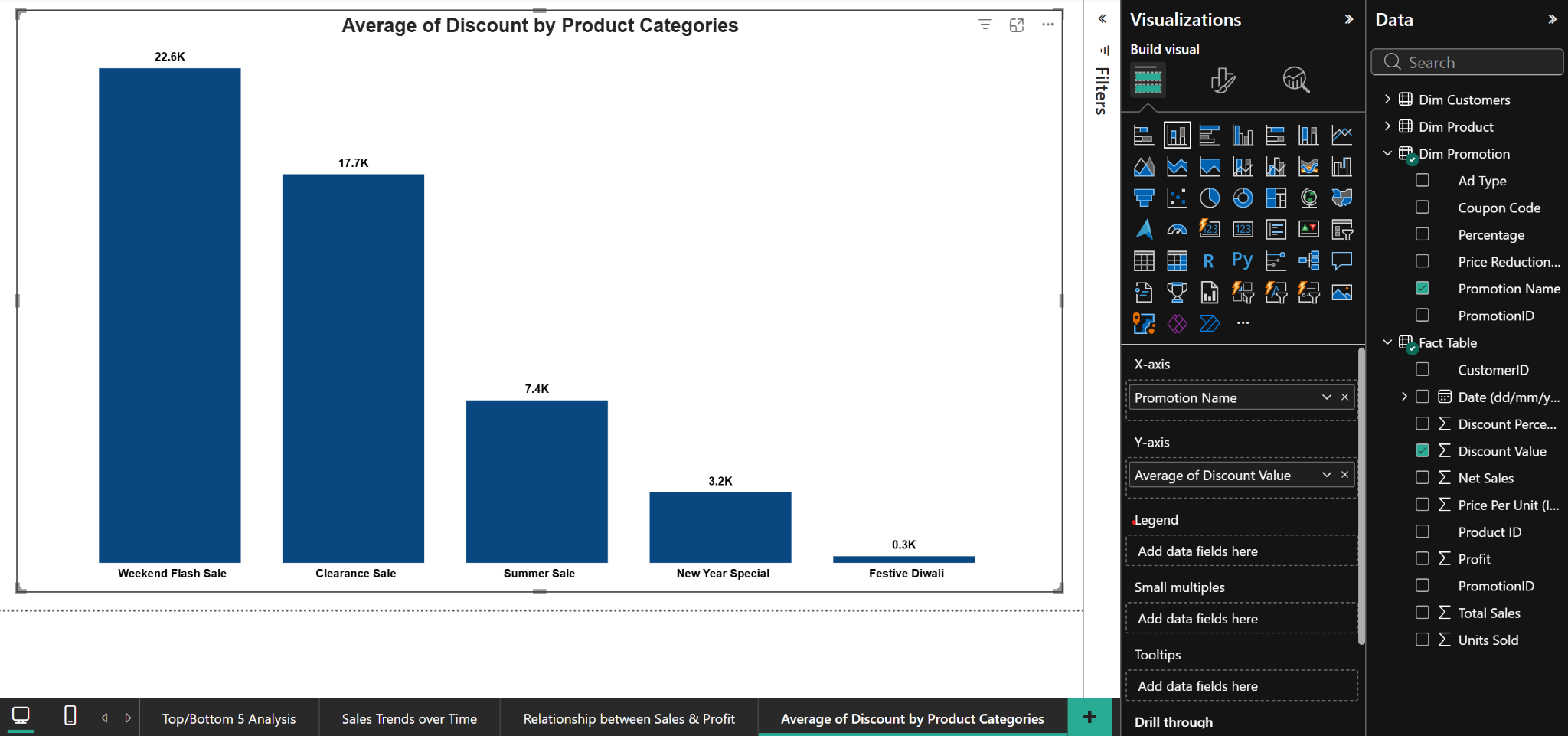
• Y-axis: `Discount Value` aggregated as \*\*Average\*\*

• Legend: not used in the current version

• Data labels: enabled to display the average discount per promotion

## Interpretation Notes

Higher values indicate promotions that, on average, grant larger monetary discounts. Because Discount Value is derived from Total Sales and Discount Percentage, promotions applied to higher-priced products can produce larger Discount Values even at similar percentage rates.



*Figure 15 – Column chart showing the average Discount Value by Promotion Name.*

# 11. Sales by City & Total Number of Orders

This page provides a geographic view of sales distribution by city and a high-level KPI card showing the total number of orders (transactions) in the dataset. It supports quick identification of high-performing cities and overall dataset volume.

## Visuals on the Page

|  |  |  |  |
| --- | --- | --- | --- |
| Visual Title | Visual Type | Field(s) | Value / Measure |
| Sales by City | Map (bubble map) | Location: Dim Customers[City] | Size: Sales metric (e.g., Sum of Total Sales or Sum of Net Sales) |
| Total Number of Orders | Card | (none) | Total Orders |

## Fields and Configuration

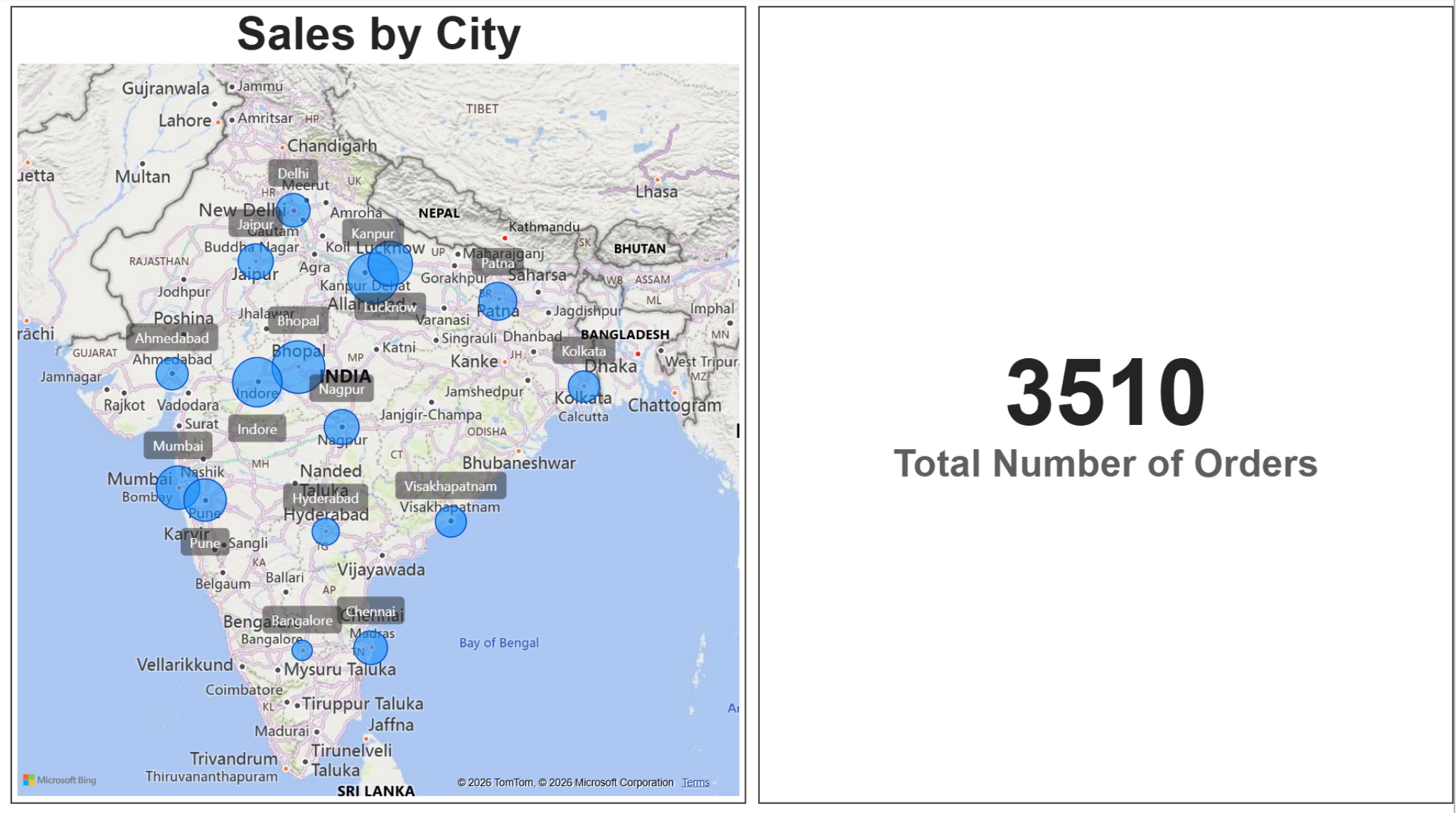
• Map visual uses \*\*City\*\* as the geographic location and a sales measure to control bubble size.

• To improve geocoding accuracy, the City column should have its \*\*Data category\*\* set to \*City\* (and optionally include Country/Region = India if ambiguity occurs).

• The KPI card shows the total number of rows in the Fact Table, which represents the total number of transaction lines.

## Interpretation Notes

Larger bubbles indicate cities contributing higher total sales. Combined with slicers (Date, Product Line, Promotion), this view can help identify where demand is concentrated and how it changes over time.



*Figure 16 – Map view of Sales by City and KPI card showing Total Number of Orders (3510).*

# 12. Report Page – Overview (High-Level Summary)

An Overview page consolidates the most important high-level visuals into a single dashboard-style layout. It is designed as the entry point of the report, enabling quick scanning of overall performance before drilling down into detailed analysis pages.

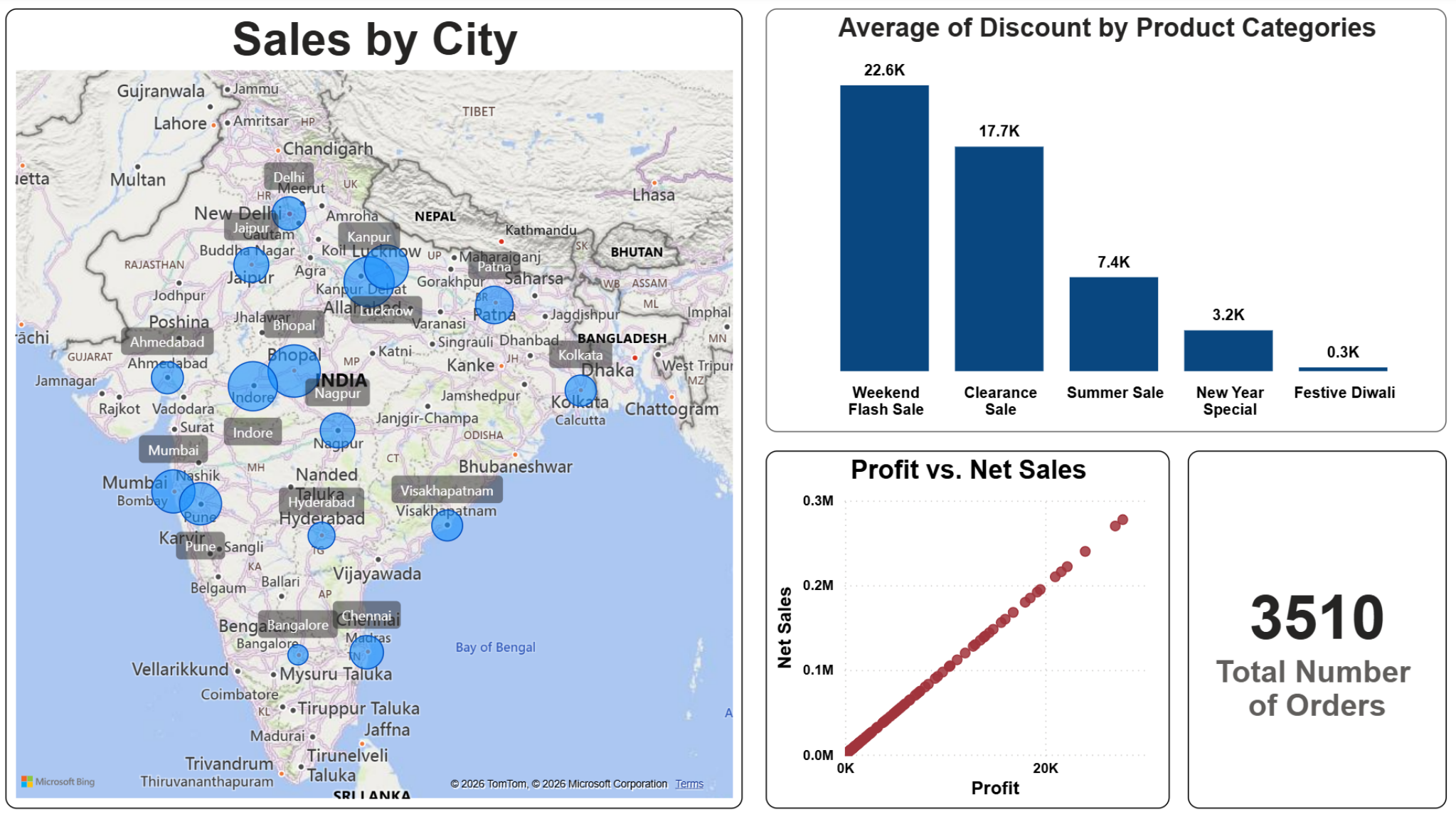
## Visuals Included

|  |  |  |
| --- | --- | --- |
| Visual | Purpose | Key Field / Metric |
| Sales by City (Map) | Geographic distribution of sales | City + Sales metric (Total/Net Sales) |
| Average Discount by Promotion (Column chart) | Compare discount intensity by campaign | Promotion Name + Avg Discount Value |
| Profit vs. Net Sales (Scatter) | Visualize revenue–profit relationship / outlier detection | Profit vs Net Sales |
| Total Number of Orders (Card) | Overall transaction volume | COUNTROWS(Fact Table) |

## Notes

• This page functions as a summary layer; filters/slicers can be added to control all visuals at once (e.g., Date, Product Line, Promotion).

• The visuals on this page are also available on their dedicated pages for deeper configuration and interpretation.



*Figure 17 – Overview page combining city sales, average discount, profit vs net sales, and total orders.*

# Report Page – User-Selected Period Comparison (Two Date Slicers)

This page is a user-driven comparison view that allows selecting \*\*two independent date ranges\*\* and comparing key KPIs side-by-side. It is implemented using \*\*two separate date tables\*\* and DAX measures that activate the appropriate relationship for the selected date range.

## What the User Can Do (Input)

• Select \*\*Date Filter 1\*\* (Period 1) using the first date slicer.

• Select \*\*Date Filter 2\*\* (Period 2) using the second date slicer.

• Compare results for the two periods across Sales, Profit, and Quantity Sold.

## KPIs Displayed (Output)

• \*\*Total Sales\*\* (Sales 1 vs Sales 2)

• \*\*Total Profit\*\* (Profit 1 vs Profit 2)

• \*\*Total Quantity Sold\*\* (Quantity 1 vs Quantity 2)

## Modeling Approach

Two date tables are used: \*\*Date Table 1\*\* (for Period 1) and \*\*Date Table 2\*\* (for Period 2). The Fact Table is filtered by one date table at a time using DAX. The second relationship is typically kept \*\*inactive\*\* and is activated only inside measures using `USERELATIONSHIP()`.

## Example DAX Measure (Period 2)

Below is an example of the DAX pattern used to calculate Net Sales for \*\*Period 2\*\*. It removes any filter coming from Date Table 1 and then applies Date Table 2 via an activated relationship:

Sum of Net Sales =  
CALCULATE(  
 SUM('Fact Table'[Net Sales]),  
 ALL('Date Table 1'),  
 USERELATIONSHIP('Date Table 2'[Date], 'Fact Table'[Date (dd/mm/yyyy)])  
)

The same pattern is repeated for the other KPIs (Total Sales, Total Profit, Quantity Sold), producing paired measures (e.g., \*Sales 1\* vs \*Sales 2\*, \*Profit 1\* vs \*Profit 2\*, etc.).

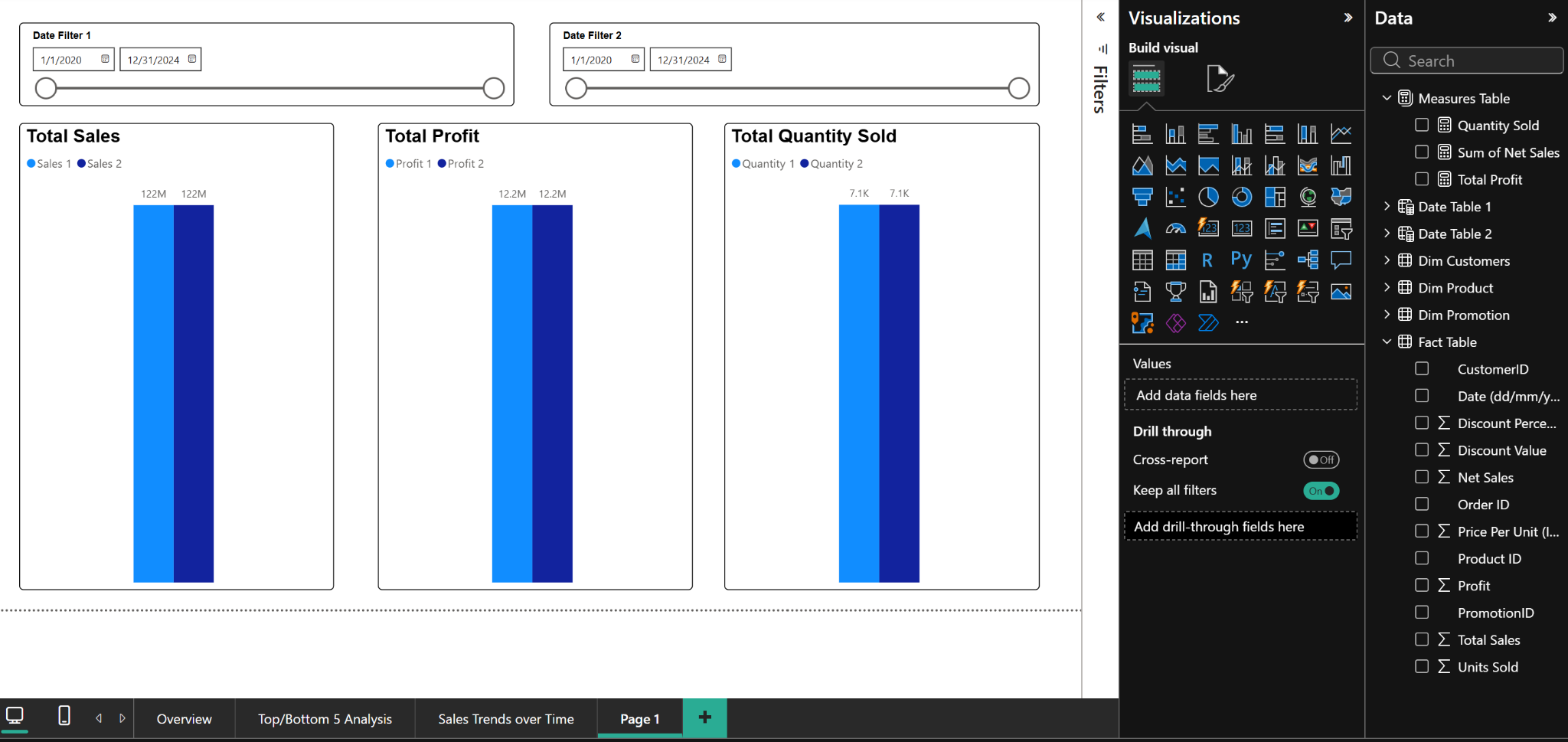
## Interpretation Notes

Because both periods are controlled by independent slicers, the page supports quick comparisons such as:

• Month-over-month / year-over-year comparisons

• Before vs after a promotion window

• Custom date ranges selected by the user



*Figure 18 – User input page with two date slicers and side-by-side KPI comparison (Period 1 vs Period 2).*