

## EXPERIMENT – 3

**Aim:** To analyze and present comprehensive insights into sales, profit, orders, profit margin, and various comparisons. It aims to provide a clear understanding of trends using Power BI.

The report objectives can be summarized as follows

- Calculate Total Sales
- Calculate Profit
- Analyze Orders
- Calculate Profit Margin
- Compare Sales by Product with Previous Year
- Compare Sales by Months with Previous Year
- Display Top 5 Cities
- Compare Profit by Channel with Previous Year
- Analyze Sales by Customer and Compare with Previous Year
- Create Slicers for Date, City, Product, and Channel

### **Solution:**

To fulfill the above given objectives we need to create business report in Power BI desktop using Sales Performance Analysis using the following steps:

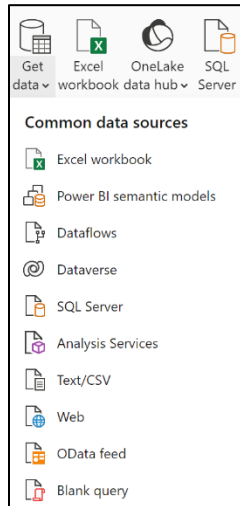
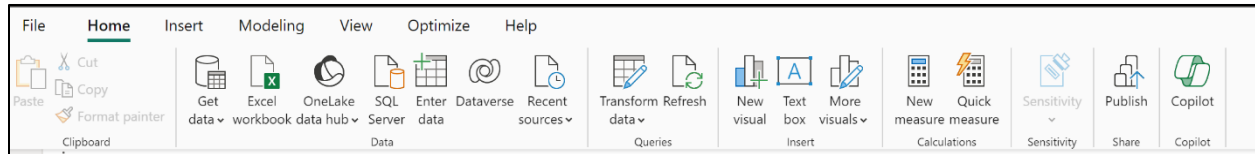
- Import data from various sources.
- Use Power Query for data cleaning and transformation.
- Create relationships between tables.
- Filter and slice your data and use drill-down capabilities for deeper analysis.
- Build calculated columns and measures using DAX.
- Create different types of charts, tables and Use slicers and filters effectively.
- Design interactive dashboards.
- Analyze the data to identify meaningful insights and make data driven decisions.

### **1. Import Data from Various Sources**

*Objective:* To bring all necessary data into Power BI for analysis.

*Steps:*

- *Open Power BI Desktop:* Launch the Power BI Desktop application.
- *Get Data:* Click on the "Get Data" button located on the Home ribbon.
- *Choose Data Source:* Select the type of data source you want to connect to (e.g., Excel, CSV, SQL Server, SharePoint, etc.). Power BI supports a wide range of data sources including cloud-based services like Azure and web-based data.



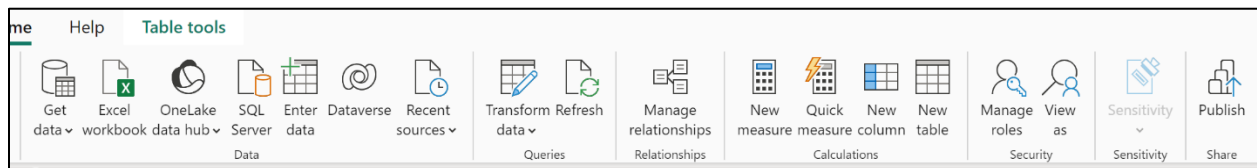
- **Connect to Data Source:** Follow the prompts to establish a connection.
- For instance, if you are importing data from an Excel file:
- Click on "Excel".
- Browse and select your Excel file.
- Click "Open".
- **Load Data:** In the Navigator window, select the tables or sheets you want to import and click "Load" to bring them into Power BI.
- Ensure your data is clean and well-structured in the source files.
- Power BI can handle large datasets efficiently, but consider the size and complexity of the data you're importing.

## 2. Use Power Query for Data Cleaning and Transformation

*Objective:* To prepare and clean the data for analysis.

*Steps:*

- **Launch Power Query Editor:** After importing your data, click on "Transform Data" to open the Power Query Editor.



*Data Cleaning:*

- **Filter Rows:** Use the filter options on column headers to include or exclude specific rows.
- **Handle Missing Values:** Replace or remove missing values using the "Replace Values" or "Remove Rows" options.

*Data Transformation:*

- **Change Data Types:** Ensure columns have the correct data types (e.g., dates, numbers, text). Right-click on the column header and select "Change Type".
- **Add Custom Columns:** Use the "Add Column" tab to create new columns using custom formulas.

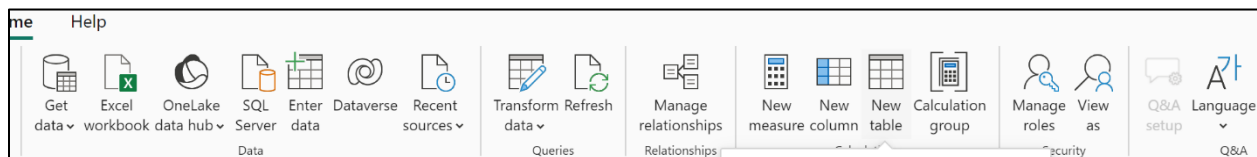
To fulfill the given objectives the following custom columns are needed

- Sales= [Order Quantity] \* [Unit Selling Price])
- Total cost= [Order Quantity] \* [unit cost]
- Profit = [Sales] - [total cost]

To create custom columns: Add required columns using Custom column of Add column

### 3. Create Relationships Between Tables

- **Objective:** To establish logical connections between different tables to enable comprehensive analysis.
- **Steps:**
  - **Go to Model View:** Click on the "Model" icon on the left sidebar to view all imported tables.



- **Create Relationships:**
  - **Drag and Drop:** Click on a field in one table and drag it to the corresponding field in another table to create a relationship.
  - **Define Relationship:** The "Manage Relationships" dialog allows you to set up relationships manually. Here, specify the primary and foreign keys.
- **Cardinality and Cross-Filtering:**
  - **Cardinality:** Define the type of relationship (one-to-many, many-to-one, many-to-many).
  - **Cross-Filtering:** Set the direction of data filtering (single or both directions).
  - Ensure that related fields have matching data types.

To accomplish our objectives we also need a Date table along with the given data.

So, let's create a DAX date table using DAX Query/Function:

```
DAX Date Table = ADDCOLUMNS(  
    CALENDARAUTO(),  
    "Year", YEAR([Date]),  
    "Quarter", "Q" & FORMAT(CEILING(MONTH([Date])/3,1), "#"),  
    "Quarter No", CEILING(MONTH([Date])/3,1),  
    "Month No", MONTH([Date]),  
    "Month Name", FORMAT([Date], "MMMM"),  
    "Month Short Name", FORMAT([Date], "MMM"),  
    "Month Short Name Plus Year", FORMAT([Date], "MMM,yy"),  
    "DateSort", FORMAT([Date], "yyyyMMdd"),  
    "Day Name", FORMAT([Date], "dddd"),  
    "Details", FORMAT([Date], "dd-MMMM-yyyy"),  
    "Day Number", DAY([Date]))
```

Certainly! Here's a concise explanation for each part of the DAX date table formula:

1. ADDCOLUMNS(CALENDARAUTO(), ...)

Purpose: Creates a date table and adds custom columns.

2. CALENDARAUTO()

Purpose: Generates a table of dates covering the date range in your data model.

3. "Year", YEAR([Date])

Purpose: Extracts the year from the date.

4. "Quarter", "Q" & FORMAT(CEILING(MONTH([Date])/3,1), "#")

Purpose: Calculates the quarter of the year (e.g., "Q1", "Q2").

5. "Quarter No", CEILING(MONTH([Date])/3,1)

Purpose: Numerical value of the quarter (1 to 4).

6. "Month No", MONTH([Date])

Purpose: Extracts the month number (1 to 12).

7. "Month Name", FORMAT([Date], "MMMM")

Purpose: Full month name (e.g., "January").

8. "Month Short Name", FORMAT([Date], "MMM")

Purpose: Abbreviated month name (e.g., "Jan").

9. "Month Short Name Plus Year", FORMAT([Date], "MMM,yy")

Purpose: Abbreviated month and year (e.g., "Jan,24").

10. "DateSort", FORMAT([Date], "yyyyMMdd")

Purpose: Sortable date string (e.g., "20240724").

11. "Day Name", FORMAT([Date], "dddd")

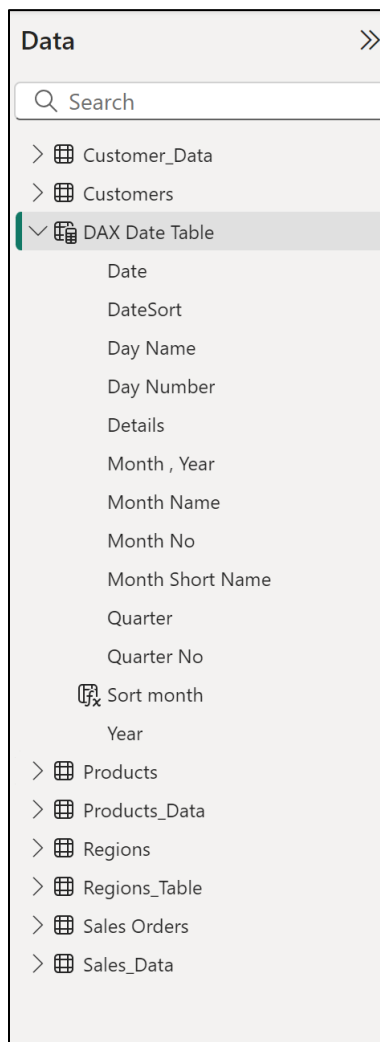
Purpose: Full day name (e.g., "Monday").

12. "Details", FORMAT([Date], "ddMMMMyyyy")

Purpose: Full date with day, month, and year (e.g., "24July2024").

13. "Day Number", DAY([Date])

Purpose: Extracts the day of the month (1 to 31).



For creating slicers for DAX table we need sorted dates. So to sort the "Month Short Name Plus Year" column in the DAX Date Table, follow these steps:

### Create a Calculated Column for Sorting:

```
Sort Month = FORMAT([Date], "yyyyMM")
```

This column will convert the date into a sortable string format "YYYYMM".

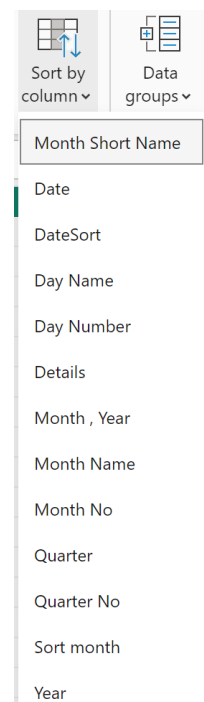
### Sort the Column:

Select the "Month Short Name Plus Year" column.

Go to the "Column Tools" tab.

Click "Sort by Column" and choose "Sort Month" (the calculated column).

This ensures that "Month Short Name Plus Year" is sorted correctly by year and month.



- Properly defined relationships are crucial for accurate data aggregation and reporting.

← New relationship

×

Select tables and columns that are related.

From table

Sales\_Data

Channel	Currency Code	Customer Na...	Delivery Regi...	Order Quantity	OrderDate	OrderNun
Wholesale	NZD	20	54	5	Thursday, Jan...	SO - 000
Wholesale	NZD	44	64	11	Thursday, Jan...	SO - 000
Wholesale	NZD	6	51	9	Tuesday, Janu...	SO - 000

To table

Products

Index	Product Name
1	Product 1
2	Product 2
3	Product 3

Cardinality

Many to one (\*:1)

Cross-filter direction

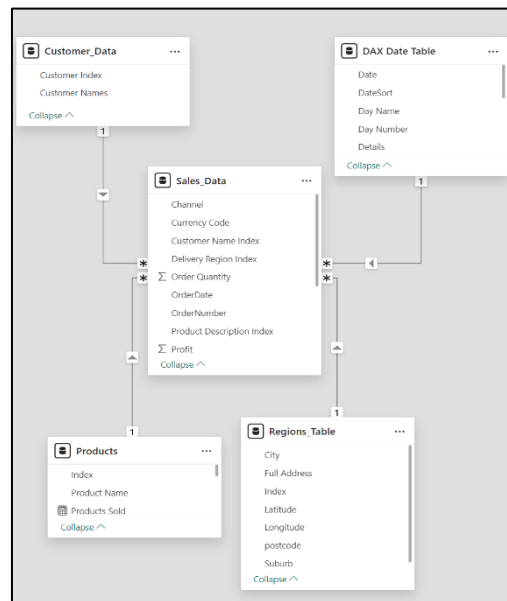
Single

☐ Make this relationship active

☐ Assume referential integrity

☐ Apply security filter in both directions

Save Cancel



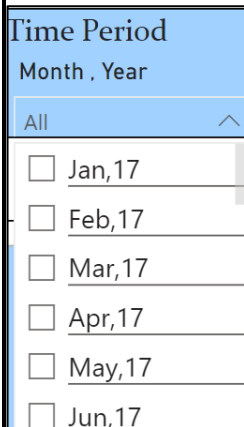
Manage relationships

×

+ New relationship Autodetect Edit Delete Filter

<input type="checkbox"/> From: table (column) ↑	Relationship	To: table (column)	Status
<input type="checkbox"/> Sales_Data (Customer Name I...	* ← 1	Customer_Data (Customer Ind...	Active ...
<input type="checkbox"/> Sales_Data (Delivery Region In...	* ← 1	Regions_Table (Index)	Active ...
<input type="checkbox"/> Sales_Data (OrderDate)	* ← 1	DAX Date Table (Date)	Active ...
<input type="checkbox"/> Sales_Data (Product Descriptio...	* ← 1	Products (Index)	Active ...

#### 4. Filter and Slice Your Data and Use Drill-Down Capabilities for Deeper Analysis



**Objective:** To enable detailed data exploration and analysis.

**Steps:**

- **Add Slicers:**

**Insert Slicer:** Go to the "Visualizations" pane, select the slicer icon, and drag it to the report canvas.

- **Configure Slicer:** Choose the field you want to use as a slicer. Adjust the slicer settings to allow single or multi-selection.

- Filters and slicers help users interact with the data dynamically.

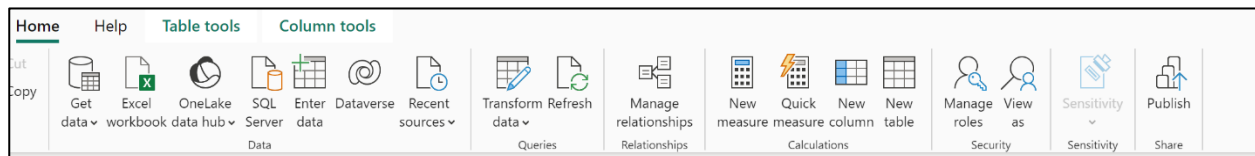
#### 5. Build Calculated Columns and Measures Using DAX

**Objective:** To perform advanced calculations and derive new insights from your data.

- DAX (Data Analysis Expressions) is a powerful language for creating complex calculations and aggregations.

**Steps:**

- **Open Data View:** Click on the "Data" icon on the left sidebar to view your tables.
- **Create Calculated Column:**
- **New Column:** Click on "New Column" in the "Home" ribbon of table view.



- **DAX Formula:** Enter a DAX formula to define the new column.  
For example, to calculate profit margin: Profit Margin = DIVIDE([Profit], [Sales]).
- **Create Measures:**
- **New Measure:** Click on "New Measure" in the "Home" ribbon of table view.
- **DAX Formula:** Define a measure using DAX.  
For example, to calculate total sales: Total Sales = SUM(Sales[Amount]).
- Calculated columns are evaluated row by row, whereas measures are aggregated calculations.

To achieve given objectives we need to find the following :

##### 1. Total Sales

total\_sales = SUM(Sales\_Data[sales])

Purpose: Calculates the total sales revenue.

**2. Previous Year Total Sales**

Sales PY = CALCULATE([total\_sales], SAMEPERIODLASTYEAR('DAX DateTable'[Date]))

Purpose: Computes total sales for the same period last year.

**3. Sales Difference (Current vs. Previous Year)**

Sales vs PY = [total\_sales] - [Sales PY]

Purpose: Shows the change in sales compared to last year.

**4. Sales Year-on-Year Percentage Change**

Sales vs py % = DIVIDE([Sales vs PY], [total\_sales], 0)

Purpose: Calculates the percentage increase or decrease in sales year-on-year.

**5. Products Sold**

Products Sold = SUM(Sales\_Data[Order Quantity])

Purpose: Total quantity of products sold.

**6. Profit**

Profit = SUM(Sales\_Data[Profit])

Purpose: Total profit from sales.

**7. Profit Last Year**

Profit LY = CALCULATE([Profit], SAMEPERIODLASTYEAR(DAX DateTable[Date]))

Purpose: Computes profit for the same period last year.

**8. Profit Difference (Current vs. Last Year)**

Profit Vs LY = [Profit] - [Profit LY]

Purpose: Shows the change in profit compared to last year.

**9. Profit Year-on-Year Percentage Change**

Profit vs LY % = [Profit Vs LY] / [Profit]

Purpose: Calculates the percentage change in profit year-on-year.

**10. Profit Margin**












Profit Margin = DIVIDE([Profit], [Sales], 0)

Purpose: Shows the profit as a percentage of sales.

**11. Total Cost**

Total Cost = SUM(Sales\_Data[Total Cost])

Purpose: Calculates the total cost associated with sales.

- ☐  Products Sold
- ☐  Profit
- ☐  Profit LY
- ☐  Profit Margin
- ☐  Profit Vs LY
- ☐  Profit vs LY %
- ☐  Sales PY
- ☐  Sales vs PY
- ☐  Sales vs py %
- ☐  Total Cost
- ☐  Total Sales



## 6. Create Different Types of Charts, Tables, and Use Slicers and Filters Effectively

*Objective:* To visualize data in various forms to communicate insights clearly.

*Steps:*

*Add Visualizations:*

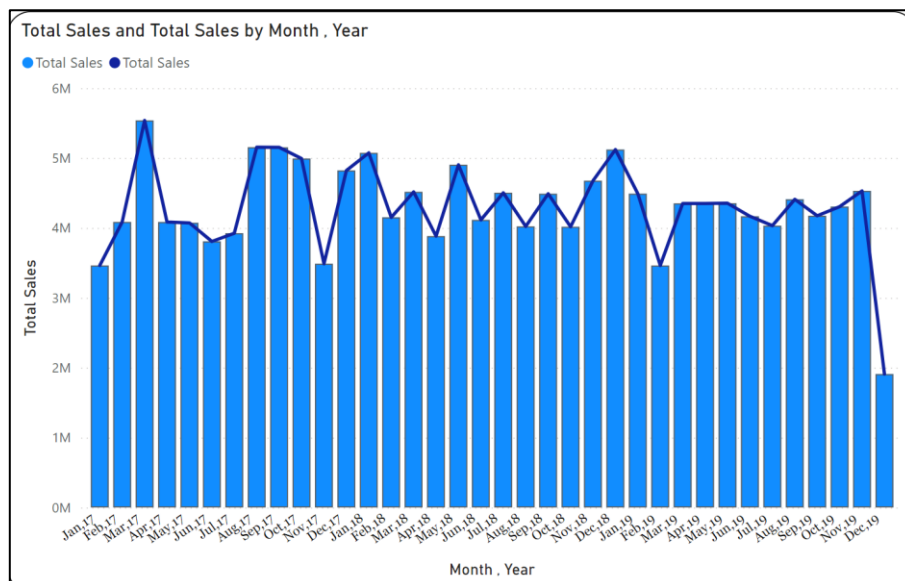
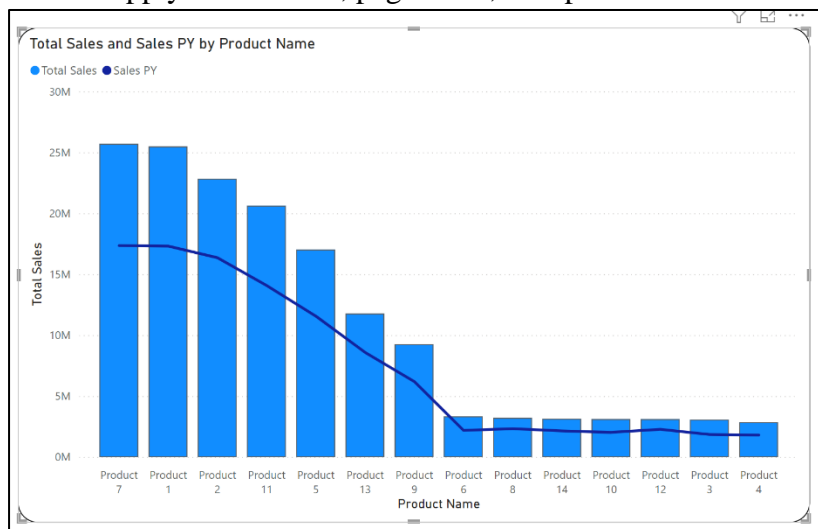
- *Select Visualization Type:* From the "Visualizations" pane, choose a chart type (e.g., bar chart, line chart, pie chart).
- *Drag Fields:* Drag and drop fields onto the visual to populate it with data.

*Customize Visuals:*

- *Format Visual:* Use the "Format" pane to customize the appearance of the visual (e.g., colors, labels, titles).
- *Add Legends and Tooltips:* Enhance visuals by adding legends and tooltips for better clarity.

*Use Slicers and Filters:*

- *Slicers:* Add slicers to allow users to filter data dynamically.
- *Filters:* Apply visual-level, page-level, or report-level filters as needed.

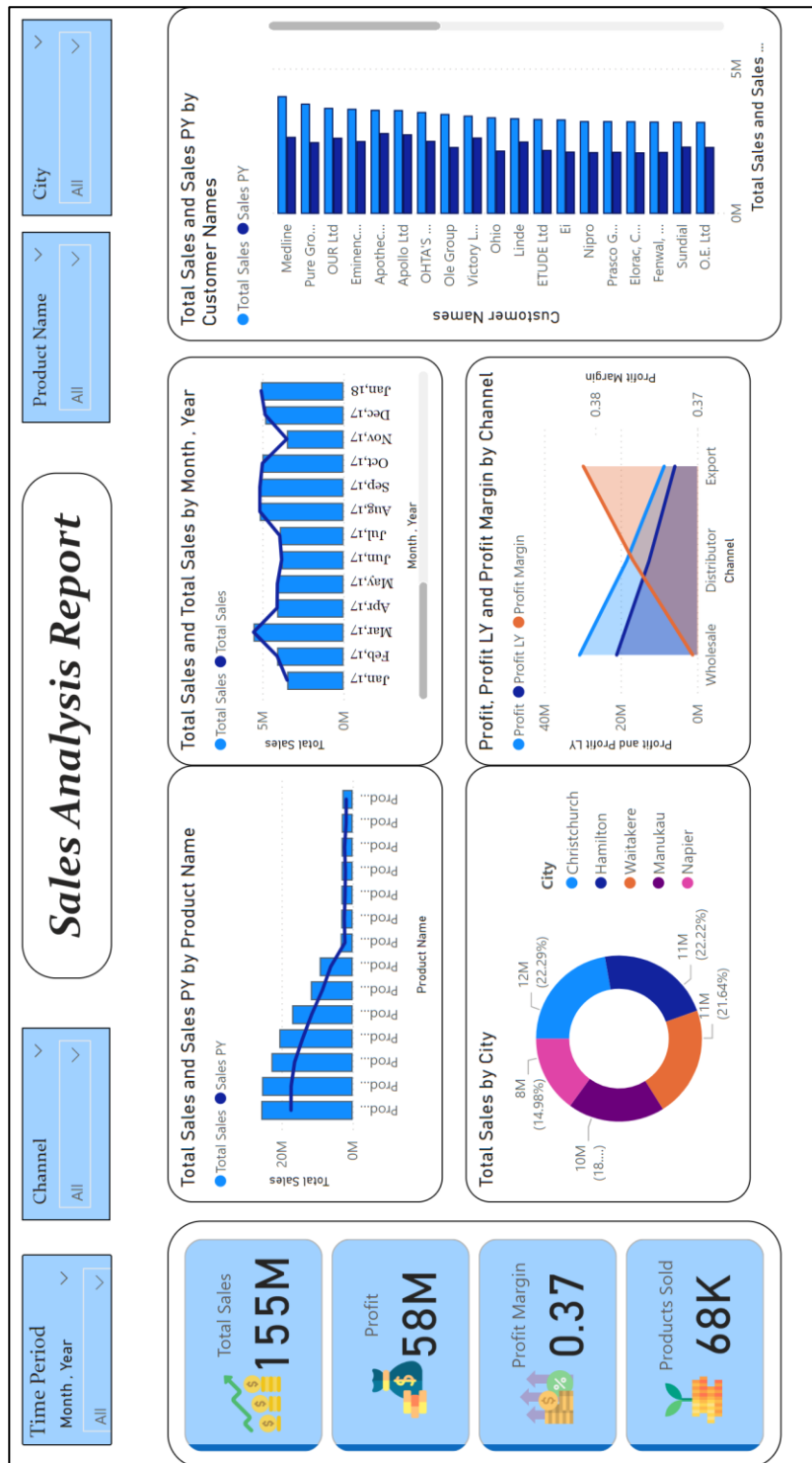


## 7. Design Interactive Dashboards

Objective: To create a user-friendly and interactive interface for data exploration.

Ensure the dashboard is intuitive and user-friendly.

Interactive elements should enhance the user experience without overwhelming them.



**8. Analyze the Data to Identify Meaningful Insights and Make Data-Driven Decisions**

Objective: To draw actionable insights from the visualized data.

Conclusion for the year 2019:

- The profit margin in the Export channel is higher
- The profit in the wholesale channel is higher
- Product -7 sales higher
- In 2017 Hamilton city having high sales, in 2018 Waitakere city having high sales and in 2019
- Christchurch city having high sales

**Result :**

Analyzed and presented comprehensive insights into sales, profit, orders, profit margin, and various comparisons.