

# Title: Microsoft Power BI Data Analytics Capstone Documentation

\*These are the steps I have followed to make this Capstone Project.

## Exercise 1: Prepare Sales Excel data

### Objective:

- Prepare the sales data worksheet to ensure accuracy before integration into Power BI.
- Understand the components of the sales data, such as individual sales per day and the corresponding gross amount.
- Implement formulas to determine the net amount by subtracting tax from the gross amount.

### Step 1: Download and open the 'Tailwind Traders Sales.xlsx' worksheet.

→ The file contains a single **Sales** worksheet.

### Step 2: Calculate Gross Revenue

- Insert a column after **Quantity Purchased** and label it **Gross Revenue**.
- Use the formula **=E2\*G2** to calculate the revenue from each product.

### Step 3: Calculate Total Tax

- Create a **Total Tax** column next to **Gross Revenue**.
- Input the formula **=F2\*G2** to calculate the tax for each product.

### Step 4: Calculate Net Revenue

- Insert a **Net Revenue** column next to **Total Tax**.
- Use the formula **=H2-I2** to determine the actual earnings post-tax for each product.
- Double-click the shortcut on cell **J2** or drag this formula down to cater to all products on your list. Repeat this action for the formulas in **H2** and **I2**.

## Exercise 2: Configure data sources

### Objective:

- Incorporate Excel data sources into Power BI and ensure accurate data types.
- Review statistics and details of the data sources to detect and rectify any discrepancies or errors.
- Identify and filter out refunded purchases to maintain the data's integrity.
- Implement a Python script to transform and refine the currency exchange data.

### Step 1: Load the Sales data

- Load the **Tailwind Traders Sales** file into Power BI and select **Transform**.
- Within Power Query, find the **OrderID** column and set the data type to **Whole Number**.
- To complete optimization, assign the following data types for the columns:
  - ◆ *Gross Product Price = Fixed Decimal Number*
  - ◆ *Tax Per Product = Fixed Decimal Number*
  - ◆ *Quantity Purchased = Whole Number*
  - ◆ *Loyalty Points = Whole Number*
  - ◆ *Stock = Whole Number*
  - ◆ *Product Category = Text*
  - ◆ *Rating = Fixed Decimal Number*
- In the **View tab**, upon selecting the **Column Quality**, **Column Distribution**, and **Column Profile** boxes, ensure the **Valid** percentage is **100%** for the **OrderID** column.

### Step 2: Load the Purchase data

- Load the **Purchases** file into Power BI and select **Transform**.
- To complete optimization, assign the following data types for the columns:
  - ◆ *PurchaseID = Whole Number*
  - ◆ *OrderID = Whole Number*
  - ◆ *Return Policy (Days) = Whole Number*
  - ◆ *Purchase Date = Date*
  - ◆ *Warranty (Months) = Whole Number*
  - ◆ *Supplier = Text*
  - ◆ *Last Visited = Date*
  - ◆ *ReturnStatus = Text*
- Select the **ReturnStatus** column and observe the **Column Quality** pane to ensure the **Valid** percentage is **100%**.
- Filter the **ReturnStatus** column to ensure that only records with **Not Returned** are visible.

### Step 3: Load the Countries data

- Load the **Countries** file into Power BI and select **Transform**.
- To complete optimization, assign the following data types for the columns:
  - ◆ *Country ID = Whole Number*
  - ◆ *Exchange ID = Whole Number*
  - ◆ *Country = Text*

### Step 4: Load the Historical currency exchange data

- Select **Get Data**, choose **Python** script, and then paste the following code into the script window in Power BI:
  - ◆ `import pandas as pd`
  - ◆ `from io import StringIO`
  - ◆ `data = """Exchange ID;ExchangeRate;Exchange Currency`
  - ◆ `1;1;USD`

```

◆ 2;0,75;GBP
◆ 3;0,85;EUR
◆ 4;3,67;AED
◆ 5;1,3;AUD""
◆ df = pd.read_csv(StringIO(data), sep=';')
◆ # Return the transformed dataframe
◆ df

```

→ Integrate this data into your Power BI report.

→ Save the Power BI project as **Tailwind Traders Report.pbix**.

→ **Note:**

◆ The Python script prepares the currency exchange data for analysis. It transforms the raw string data into a structured format that can be easily integrated with other datasets within Power BI. The core script elements are as follows:

→ The **pandas** data analysis library is used for manipulating and analyzing data.

→ **StringIO** is a module that lets you read and write strings like files.

→ **pd.read\_csv()** is a pandas function that reads a CSV file into a DataFrame.

## Exercise 3: Design and develop the data model

### Objective:

- Construct a snowflake schema tailored for the data model.
- Establish and specify the relationships between various tables in the data model.
- Define a dedicated calendar table for temporal data analysis and reporting.
- Synthesize DAX data into a consolidated US Dollar table encompassing sales and profit metrics.

### Step 1: Create a relationship between the Countries and Exchange Data tables

- Create a relationship between the **Countries** and **Exchange** Data tables on the **Exchange ID** field.
- Set the **Cardinality** to **One to One (1:1)**
- Set the **Cross filter direction** to **Both** to be bi-directional.
- Ensure the **Make this relationship active** checkbox is selected.
- Inspect the relationship arrow in the Model View to ensure the arrows point in both directions and display a **1:1** symbol on either end of the connector.

### Step 2: Create a relationship between the Sales and Countries tables

- Create a relationship on the **Country ID** field between the **Sales** and **Countries** tables.
- Set the **Cardinality** to **Many to One (1:1)**.
- Set the **Cross-filter direction** to **Both** so that it's bi-directional.
- Ensuring the **Make this relationship active** checkbox is selected.
- Inspect the relationship arrow in the Model View to ensure the arrows point in both directions and display a **\*:1** symbol on either end of the connector.

### **Step 3: Create a relationship between the Purchases and Sales tables**

- Create a relationship on the **OrderID** field between the **Purchases** and **Sales** tables.
- Set the **Cardinality** to **One to One (1:1)**.
- Set the **Cross filter direction** to **Both** to be bi-directional.
- Ensure the **Make this relationship active** checkbox is selected.
- Inspect the relationship arrow in the Model View to ensure the arrows point in both directions and display a **1:1** symbol on either end of the connector.

### **Step 4: Configure the Calendar table**

- Select **New Table** and add the following DAX code to create a new **Calendar table**:

```

◆ CalendarTable =
◆ ADDCOLUMNS(
◆ CALENDAR(DATE(2020, 1, 1), DATE(2023, 12, 31)),
◆ "Year", YEAR([Date]),
◆ "Month Number", MONTH([Date]),
◆ "Month", FORMAT([Date], "MMMM"),
◆ "Quarter", QUARTER([Date]),
◆ "Weekday", WEEKDAY([Date]),
◆ "Day", DAY([Date]))

```

### **Step 5: Create a relationship between the Calendar and Purchases tables**

- Create a relationship on the **Date** field between the **Calendar** and on **Purchase Date** in the **Purchases** table.
- Set the **Cardinality** to **Many to One (1:1)**.
- Ensure the **Make this relationship active** checkbox is selected.

- Inspect the relationship arrow in the Model View to ensure the arrows point in both directions and display a **\*:1** symbol on either end of the connector.

#### **Step 6: Create a Sales in USD calculated table**

- Select **New Table** and add the following DAX code to create a new calculated table:
- ◆ Sales in USD =
  - ◆ ADDCOLUMNS(
    - ◆ Sales,
    - ◆ "Country Name", RELATED(Countries[Country]),
    - ◆ "Exchange Rate", RELATED('Exchange Data'[Exchange Rate]),
    - ◆ "Exchange Currency", RELATED('Exchange Data'[Exchange Currency]),
    - ◆ "Gross Revenue USD", [Gross Revenue] \* RELATED('Exchange Data'[Exchange Rate]),
    - ◆ "Net Revenue USD", [Net Revenue] \* RELATED('Exchange Data'[Exchange Rate]),
    - ◆ "Total Tax USD", [Total Tax] \* RELATED('Exchange Data'[Exchange Rate]))

#### **Step 7: Create a relationship between the Sales in USD and Sales tables**

- Create a relationship between the **Sales in USD** and **Sales** tables on the **Order ID** field.
- Set the **Cardinality** to **Many to One (1:1)**.
- Ensuring the **Make this relationship active** checkbox is selected.
- Inspect the relationship arrow in the Model View to ensure the arrows point in both directions and display a **1:1** symbol on either end of the connector.

### **Exercise 4: Configure aggregations using DAX**

#### **Objective:**

- Create time-based summaries for displaying quarterly, annual, and year-to-date profit data.
- Determine median sales volume to assess Tailwind Traders' performance stability.
- Utilize the **Performance Analyzer** tool to enhance report generation and ensure fast loading times.

### **Step 1: Calculate Yearly Profit margin**

- Create a new measure for the **Sales in USD** table.
- In the formula bar, create a new measure using a **DAX** function that represents the yearly profit margin. This margin should be calculated by dividing total profit by net revenue within the **Sales in USD** table.
- In the **Fields pane**, select the new measure and change its format to **Percentage in the Properties pane**.
- **Tip:**
  - ◆ Consider using **DIVIDE** and **SUM** functions to create this ratio of profit to revenue.
- **DAX Code:**
  - ◆ Yearly Profit Margin =
  - ◆ DIVIDE(
    - ◆ SUM('Sales in USD'[Profit USD]),
    - ◆ SUM('Sales in USD'[Net Revenue USD]))

### **Step 2: Calculate Quarterly Profit**

- Right-click on the **Sales in USD** table in the **Fields** pane and choose **New Measure**.
- In the formula bar, create a new measure for quarterly profit using the **Profit in USD** column. You'll need a function that aggregates data until the end of the current quarter, referencing both the profit values and a calendar table.
- Format the new measure as a **Percentage**.
- **Tip:**
  - ◆ Consider using the **DATESQTD** function with your profit values to aggregate data by quarter.
- **DAX Code:**
  - ◆ Quarterly Profit Margin =
  - ◆ CALCULATE(
    - ◆ [Yearly Profit Margin],
    - ◆ DATESQTD('CalendarTable'[Date]))

### **Step 3: Calculate Year-to-Date Profit**

- Right-click on the **Sales in USD** table in the **Fields** pane and select **New Measure**.

- In the formula bar, create a new measure for the year-to-date profit using the **Profit in USD** column. You'll need a function that aggregates profit data from the start of the year up to the current date.
- Format the new measure as a **Percentage**.
- **Tip:**
  - ◆ Consider using **TOTALYTD** with your profit values and calendar table to calculate running totals from the start of each year.
- **DAX Code:**
  - ◆ YTD Profit Margin =
  - ◆ `TOTALYTD([Yearly Profit Margin],'CalendarTable'[Date])`

#### **Step 4: Calculate Median Sales**

- Right-click on the **Sales in USD** table in the **Fields pane** and choose **New Measure**.
- In the formula bar, create a new measure to represent the median sales based on the **Gross Revenue USD** column. Consider which statistical functions in **DAX** can help you find the middle value of your sales data.
- **Tip:**
  - ◆ Consider using the **MEDIAN** function to find the central value in your **Gross Revenue USD** data. The median separates the higher and lower half of a data sample.
- **DAX Code:**
  - ◆ `Median Sales = MEDIAN('Sales in USD'[Gross Revenue USD])`

#### **Step 5: Access the Performance Analyzer**

- Find and select the **Performance Analyzer** option within the **View** tab.
- Create an empty **Card** visual and drag the **Yearly Profit Margin** field to the **Fields well**. Repeat this process for the **Median Sales**, **Quarterly Profit**, and **YTD Profit**.
- Begin recording the performance of the card visuals using the Performance Analyzer's recording feature.
- Refresh your reports to test their performance.
- Select the **plus (+)** symbol next to each **Card** to open up the details.
- Observe the list of all visual items in your report and their respective load times. Ensure the **DAX** query time of visual items is < 200ms and note any slow-loading visuals.
- Select **Stop** and remove all **Card** visuals, resulting in a blank Canvas.

→ Save your report.

## Exercise 5: Create a Sales report

### Objective:

- Create different kinds of charts to display sales data.
- Display important sales metrics using cards and KPIs.

### Step 1: Create a Sales Overview report

- Open the **Tailwind Traders Report.pbix** Power BI file.
- Rename the report from **Page 1** to **Sales Overview**.

### Step 2: Create a bar chart for loyalty points by country

- From the **Visualizations pane**, select the clustered bar chart.
- Open the **Data Fields** pane.
- Select the **Sales in USD** table to expand it and view its fields.
- Add data from the **Sales in USD** table:
  - ◆ Drag **Country** to the Y-axis field
  - ◆ Drag **Loyalty Points** to the X-axis field
- Resize and position the chart to the left side of the canvas.
- Set the title to **"Loyalty Points by Country"**.
- Enable data labels.
- Note the country with the highest loyalty points value.
- **Tip:**
  - ◆ You may need to sort the chart. To sort the bars in descending order to immediately highlight top performers, select the "More options" (...) menu on the chart and select "Sort axis" then "Country Name - Ascending"

### Step 3: Create a column chart for quantity sold by product

- From the **Visualizations pane**, select the clustered column chart.
- Add data from the **Sales in USD** table:



- ◆ Drag **Product** to the X-axis field
- ◆ Drag **Quantity** to the Y-axis field
- Configure the chart. To start, resize and position the chart to the right of the **Loyalty Points chart**.
- Set the title to **"Quantity Sold by Product"**.
- Enable data labels.

#### **Step 4: Create a pie chart for median sales distribution by country**

- From the **Visualizations pane**, select the pie chart.
- Add data from the **Sales in USD** table:
  - ◆ Drag **Country** to the **Legend field**
  - ◆ Drag **Median Sales** to the **Values field**
- Configure the chart. To start, resize and position the chart below the **Loyalty Points chart**.
- Set the title to **"Median Sales Distribution by Country"**.
- Enable data labels.
- Sort the data in ascending order.

#### **Step 5: Create a line chart for median sales over time**

- From the **Visualizations pane**, select the line chart.
- Add data from the **Sales in USD** table:
  - ◆ Drag **Date** to the X-axis field
  - ◆ Drag **Median Sales** to the Y-axis field
- Configure the chart. To start, resize and position the chart below the **Quantity Sold by Product column chart**.
- Set the title to **"Median Sales Over Time"**.
- Enable data labels.
- Enable a Trend line.
- **Tip:**
  - ◆ You can set a **Trend line** in the **Analytics tab** of the **Visualizations pane**.

#### **Step 6: Create cards to visualize your measures**

- From the **Visualizations pane**, select the card visualization.
- Add data from the **Sales in USD** table:

- ◆ Drag **Stock** to the **Fields** section
- Create a second card and add:
  - ◆ Drag **Quantity Purchased** to the **Fields** section
- Create a third card and add:
  - ◆ Drag **Median Sales** to the **Fields** section
- Set the titles as **Stock**, **Quantity Purchased**, and **Median Sales**.
- Position the **Stock** and **Quantity Purchased** cards above the **Loyalty Points by Country bar chart**.
- Position the **Median Sales card** above the **Quantity Sold by Product column chart**.
- **Tip:**
  - ◆ Keep your card designs simple for clear and immediate data presentation.

#### **Step 7: Add a slicer to the report**

- From the **Visualizations pane**, select the slicer visualization.
- Add data from the **Sales in USD** table:
- Drag **Country Name** to the **Fields** section
- Position the slicer above the **Quantity Sold by Product** column chart.
- Save your report.

## **Exercise 5: Create a Profit report**

### **Objective:**

- Create different kinds of charts to display profit data.
- Display important profit metrics using cards and KPIs.

#### **Step 1: Create a Profit Overview report**

- Open your **Sales Overview** report.
- Create a new page in your existing **Sales Overview** report and name it **Profit Overview**.

#### **Step 2: Create a bar chart for Net Revenue by Product**

- From the **Visualizations pane**, select the clustered bar chart.
- Open the **Data Fields pane**.
- Select the **Sales in USD** table to expand it and view its fields.
- Add data from the Sales in USD table:
  - ◆ Drag **Product Name** on the X-Axis
  - ◆ Drag **Net Revenue USD** on the Y-Axis.
- Resize and position the chart to the left side of the canvas.
- Set the title to **"Net Revenue by Product"**.
- Enable data labels.
- Note the product with the highest **Net Revenue** value.
- Sort the data in descending order.
- **Tip:**
  - ◆ You can customize your chart using the **Format** tab to change colors, add titles, and adjust scales.

### **Step 3: Create a donut chart for Yearly Profit Margin by Country**

- From the **Visualizations pane**, select the donut chart.
- Add data from the **Sales in USD** table:
  - ◆ Drag **Country Name** in the **Legend** area.
  - ◆ Drag **Yearly Profit Margin** in the **Values** area.
  - ◆ Configure the chart. To start, resize and position the chart to the right of the **Net Revenue by Product chart**.
- Set the title to **"Yearly Profit Margin by Country"**.
- Enable detailed labels for **Category**, percent the **Percent** of total category.

### **Step 4: Create an area chart for Yearly Profit Margin over Time**

- From the **Visualizations pane**, select the area chart.
- Add data from the **CalendarTable** and **Sales in USD** table:
  - ◆ Drag **Date** to the X-axis field
  - ◆ Drag **Yearly Profit Margin** to the Y-axis
  - ◆ Configure the chart. To start, resize and position the chart below the **Net Revenue by Product chart**.
- Set the title to **"Yearly Profit Margin Over Time"**.
- Enable data labels.

### **Step 5: Create cards to visualize your measures**

- Create cards that visualize the following measures:
  - ◆ **YTD Profit**
  - ◆ **Net Revenue USD**
- Position the cards above the **Net Revenue by Product bar chart**.
- **Tip:**
  - ◆ Keep the card design simple and legible for quick data reference.

### **Step 6: Create a KPI for Gross Revenue USD**

- Create a KPI for **Gross Revenue USD**.
- Configure the KPI as follows:
  - ◆ Display the **Gross Revenue USD** in the **value area**
  - ◆ Display the **Date** in the **Trend Axis area**.
- Resize and position the KPI next to the **Net Revenue USD** card.

### **Step 7: Add a slicer to the report**

- From the **Visualizations pane**, select the slicer visualization.
- Add data from the **CalendarTable** table:
  - ◆ Drag **Data** to the **Fields** section
- Position the slicer next to the **Gross Revenue KPI**.
- Save your report.
- Publish your report to Power BI Service.