### **Step 1: Load the Data into Power BI**

1. Open Power BI Desktop and select Home > Get Data.
2. Choose Excel, navigate to the .xlsx file, and load the SalesData and CustomerDetails sheets.
3. Click on Transform Data to open Power Query Editor.

### **Step 2: Data Cleaning in Power Query Editor**

#### **1. Remove Duplicates**

* In the SalesData table:
  + Select the OrderID column, then right-click and choose Remove Duplicates.
* In the CustomerDetails table:
  + Select the CustomerID column and remove duplicates in a similar manner.

#### **2. Handle Missing Values**

* **In SalesData:**
  + Select the Quantity column, go to Transform > Replace Values, and replace null with 0.
  + For the Price column, use Transform > Replace Values to replace null with the average or median value of the column. This can be calculated using Add Column > Statistics > Average.
  + For OrderDate, use Transform > Replace Values to replace null with a placeholder like "Unknown" or an earliest date.
  + For Country and SalesRep, use Transform > Replace Values to replace null with "Unknown."
* **In CustomerDetails:**
  + Replace null values in the Name, Email, and Country columns with "Unknown."
  + For PhoneNumber, replace null values with "N/A".

#### **3. Remove NULL and Empty Rows**

* In each table, select the columns critical for analysis (e.g., OrderID in SalesData, CustomerID in CustomerDetails).
* Use Remove Rows > Remove Blank Rows to eliminate rows where these columns are empty.

#### **4. Trim Whitespace**

* Select text columns (e.g., CustomerName, Product, SalesRep) and use Transform > Format > Trim to remove leading and trailing spaces.

#### **5. Data Type Corrections**

* Convert columns to appropriate data types:
  + Select Quantity and set the data type to Whole Number.
  + Set Price to Decimal Number.
  + Set OrderDate to Date.

#### **6. Standardize Text Case**

* Select text columns (e.g., Country, CustomerName) and use Transform > Format > Capitalize Each Word to ensure consistency.

#### **7. Validate Data**

* In the Quantity column, use Add Column > Conditional Column to check for non-numeric values and flag them if necessary.

### **Step 3: Data Transformation in Power Query Editor**

#### **1. Column Splitting**

* Select the OrderDate column, then use Add Column > Date > Year, Month, and Day to create new columns for each component.

#### **2. Creating New Columns**

Add a new column TotalValue by using Add Column > Custom Column.

Enter the formula:  
[Quantity] \* [Price]

#### **3. Merge Queries**

* Select Home > Merge Queries.
* Merge SalesData with CustomerDetails using CustomerName as the matching field. Choose the join type as needed (e.g., Left Outer).

#### **4. Conditional Columns**

* Create a new column OrderStatus using Add Column > Conditional Column:
  + If OrderDate and SalesRep are not null, set the status to "Complete."
  + Otherwise, set it to "Incomplete."

#### **5. Replace Values**

* To standardize country names, select the Country column, use Transform > Replace Values, and replace variations like "U.S.A" with "USA".

#### **6. Sort and Filter Data**

* Sort the SalesData by OrderDate or TotalValue using the Sort Ascending/Descending options in the column headers.
* To filter out unnecessary rows, use the filter dropdowns in the column headers.

#### **7. Remove Unnecessary Columns**

* Select columns that are not needed (e.g., PhoneNumber if not relevant) and use Remove Columns.

#### **8. Group Data**

* To aggregate data, select columns (e.g., Product, Country) and use Home > Group By. Choose the aggregation method (e.g., sum of Quantity or TotalValue).

#### **9. Format Data**

* Apply number formatting (e.g., currency for Price and TotalValue) using Transform > Data Type.

### **Step 4: Advanced Transformations (Optional)**

#### **1. Custom Calculations**

To create a DiscountedPrice column, use Add Column > Custom Column with a formula like:  
  
if [TotalValue] > 1000 then [TotalValue] \* 0.9 else [TotalValue]

#### **2. Date Transformations**

* Create additional date-based columns using Add Column > Date transformations, such as Quarter or custom financial periods.

### 

### 

### **Step 5: Apply Changes**

1. Once all transformations are complete, click Close & Apply in the Power Query Editor to load the cleaned and transformed data into Power BI for analysis.

### **Final Notes**

* **Refresh Data:** If the source data changes, refresh the data in Power BI, and all cleaning and transformation steps will be reapplied.
* **Document Steps:** Power Query automatically documents each step. You can modify, reorder, or delete steps in the Applied Steps pane.

**Solution for tasks for : Filtering and Managing Parameters**

#### **Step 1: Open Power Query Editor**

1. In Power BI Desktop, load the .xlsx file and open the SalesData and CustomerDetails sheets.
2. Go to Home > Transform Data to open the Power Query Editor.

#### **Step 2: Filtering Rows Based on Conditions**

1. **Date Filtering:**
   * In the SalesData table, click on the OrderDate column header.
   * Select Date Filters > After.
   * Choose a specific date (e.g., "2024-01-01") to filter orders that occurred after this date.
2. **Numeric Filtering:**
   * In the SalesData table, click on the TotalValue column (create this column first as described earlier).
   * Select Number Filters > Is greater than.
   * Enter 0 to remove rows where TotalValue is zero or negative.
3. **Text Filtering:**
   * In the Country column, click the filter icon.
   * Manually select the countries you want to include (e.g., USA, Canada).

#### **Step 3: Creating Parameters**

1. **Create a Parameter for Date Filtering:**
   * In the Power Query Editor, go to Home > Manage Parameters > New Parameter.
   * **Name:** Set the name to "MinOrderDate."
   * **Data Type:** Select Date/Time.
   * **Default Value:** Choose a default date, e.g., "2024-01-01."
   * **Allowed Values:** Set this to "Any Value" or define a list of specific dates.
2. **Create a Parameter for Country Filtering:**
   * Go to Home > Manage Parameters > New Parameter.
   * **Name:** Set the name to "CountryFilter."
   * **Data Type:** Choose Text.
   * **Allowed Values:** Enter a list of possible countries (e.g., USA, Canada, UK).
   * **Default Value:** Set the default to one of the countries (e.g., "USA").

#### **Step 4: Using Parameters in Data Filtering**

1. **Apply Date Parameter to the Filter:**
   * In the SalesData table, select the OrderDate column.
   * Click on Date Filters > After.
   * Select "Parameter" and choose "MinOrderDate" as the reference value.
   * This will filter rows based on the date specified in the "MinOrderDate" parameter.
2. **Apply Country Parameter to the Filter:**
   * In the Country column, click the filter icon.
   * Select Text Filters > Equals.
   * Choose the "CountryFilter" parameter.
   * This filter will dynamically change based on the value set in the "CountryFilter" parameter.

#### **Step 5: Testing and Modifying Parameters**

1. **Test Parameters:**
   * In Power Query Editor, change the parameter values:
     + Go to Manage Parameters.
     + Modify the "MinOrderDate" to a new date (e.g., "2024-03-01").
     + Change the "CountryFilter" to a different country (e.g., "Canada").
   * Click Close & Apply to see how the data updates in Power BI based on the new parameter values.
2. **Use Parameters in Power BI Reports:**
   * After applying the changes, the filtered data will reflect in your Power BI report.
   * You can further create slicers or dropdowns in the Power BI report to let users interact with parameters for a dynamic report experience.

#### **Step 6: Dynamic Filtering with Parameters in Reports (Optional)**

1. In the report view of Power BI Desktop, you can add slicers or dropdown menus that reference the parameters, allowing users to adjust the data view directly in the report.

### **Final Notes**

* Parameters enable a flexible and dynamic way of filtering data, making your Power BI reports more interactive.
* Ensure that the parameters are well-defined and provide useful options for data exploration.
* When managing parameters, you can quickly adapt the data transformations without altering the underlying queries, saving time and making reports more user-friendly.