1. **Data extraction**:

Data was extracted using importing methods in PowerBI for the Excel, CSV and text filetypes.

2. **Data Management:**

The loaded data tables are inspected for the existing and missing connections between them. Any missing connections are established by creating relationships between the tables by their common variables. The missing connection between Salespeople and customer tables is established using the sales ID column. Any missing connections post the transformations or addition of new tables are subsequently created.

3. **Data Transformation:**

The following transformations are done to prepare the data for analysis. The data is first checked for missing values. The missing values in the PRODUCT_COLOUR column of the products table and CONTACT_EMAIL column of the Logistics table are replaced with null as the removal of these rows might result in loss of necessary information. Next, the variables are checked for appropriate data types. PRICE_PER_UNIT (Products), PRODUCT_PRICE (Transactions) and PRODUCT COST (Transactions) are converted from whole numbers to decimal type since prices can have decimal values.

For the creation of visualisations the following transformations are performed:

Transaction Table:

- To calculate the profits from each transaction, a custom column PRODUCT_PROFIT is created in the Transactions table by using the formula PRODUCT_PRICE-PRODUCT_COST. Its type is set to decimal.
- A new column “month” is created in the Transactions table to extract the month from the transaction date. For the creation of visualisation to show the impact of sustainability commitment, a new column “pre-july” is created using the DAX formula pre-july = IF(Transactions[month]<=6, “pre-july”, “post-july”).
- A new column “continent” is added to the transactions table, to differentiate shipment countries belonging to the UK and Europe.

Products Table:

- A new column ‘Product_Line’ is added using a conditional column to categorise the products based on their type. It is derived from the PRODUCT_NAME column without the colour information of the product.

Product_Sales_details:

- To create a new table ‘Product_Sales_Details’ with complete details of a product, the Transactions table is duplicated. Only necessary columns (PRODUCT_ID, QUANTITY, PRODUCT_PRICE, PRODUCT_COST and PRODUCT_PROFIT) are retained. The data is grouped by PRODUCT_ID to ensure only one record for each

product. A CATEGORY column is added based on the corresponding categories in the products table. The PRODUCT_SERIES column is added based on the corresponding Product_line column in the products table using a DAX formula.

- PRODUCT_SERIES = Related(Products[Product_line])
- CATEGORY = Related(Products[PRODUCT_CATEGORY])

- To identify the most profitable products, the profit margin is calculated. Since the products table has missing and negative values including some irregularities (very high value compared to other products in the same category), for PRICE_PER_UNIT and COST, the following columns are created in the Product_Sales_Details table created earlier:
  - PER_UNIT_PRICE = PRODUCT_PRICE/QUANTITY
  - PER_UNIT_COST = PRODUCT_COST/QUANTITY
  - PER_UNIT_PROFIT = PRODUCT_PROFIT/QUANTITY
  - PROFIT_MARGIN(%) = ([PRODUCT_PRICE-PRODUCT_COST]/PRODUCT_PRICE)* 100
- A new column “rank” is created in Product_Sales_Details to rank top-selling products in each category based on quantity sold. This is calculated as

  rank =
  RANKX(FILTER('Product_Sales_Details',Product_Sales_Details[CATEGORY]=EARLIER(Product_Sales_Details[CATEGORY])), [QUANTITY],,DESC, Dense)

Thus, all the required transformations and management are performed to enable the creation of desired visualisations.