**Project Overview**

This project analyzes supplier quality by examining defect quantities and downtime across various materials. Using Python, we Cleaned our data by removing duplicates and dealing with nulls to enhance overall analysis performance and efficiency.

**Project data**

Vendor - Contains information about different vendors.

Plant - Details the plants associated with the vendors.

Defected Items - Information about items that were defected.

Material Type - Different types of materials involved.

Defects - A record of the defects.

Defect Type - Categories or types of defects.

Category - Likely involves categorization of defects or vendors.

**Project Background**

This project focuses on building a data analytics dashboard to streamline defect analysis and optimize production quality. The dashboard will integrate the following key data elements:

Vendors & Plants: Tracks vendors through unique Vendor IDs and plant locations with Plant IDs, enabling geographic performance comparisons and vendor benchmarking.

Defected Items: Logs defects by Date, Sub Category ID, Plant ID, Vendor ID, Material ID, Defect Type ID, and Defect ID, along with defect quantity and downtime, providing insights into production disruptions.

Material Types: Defines materials by Material Type ID, allowing for trend analysis and the identification of materials that contribute to higher defect rates.

Defect Details: Classifies defects using Defect IDs and Defect Types to categorize issues as "rejected," "impact," or "no impact" for prioritization and corrective actions.

Category System: Groups defects into Categories and Sub Categories to uncover broader patterns and drive continuous quality improvement.

Data Cleaning using python :

1) Upload file :

We Start by importing pandas library and reading data :

from google.colab import files

# Upload file

uploaded = files.upload()

import pandas as pd

df = pd.read\_csv(list(uploaded.keys())[0])

2) Read Data :

import pandas as pd

df = pd.read\_csv('Defected Items.csv')

3) Show Original Data :

print("Show original data:")

print(df)

4) Show ‘Downtime min’ Data :

empty\_downtime\_rows = df[df['Downtime min'].isnull()]

print("\n Show original data 'Downtime min':")

print(empty\_downtime\_rows)

5) Deleting empty values in 'Downtime min' and show data after cleaning :

df\_cleaned = df.dropna(subset=['Downtime min'])

print("\n Show data after deletion 'Downtime min':")

print(df\_cleaned)

6) Save data after modification :

df\_cleaned.to\_csv('Defected\_Items\_Cleaned.csv', index=False)

7) Converting Date Column to Date Format and Saving Updated Data to CSV :

df['Date'] = pd.to\_datetime(df['Date']).dt.date

print("\n Display data after modification:")

print(df)

df.to\_csv('Defected\_Items\_Updated.csv', index=False)

8) Removing Duplicates and Displaying Updated Data :

df\_unique = df.drop\_duplicates()

print("\n # Display data after modification")

print(df\_unique)

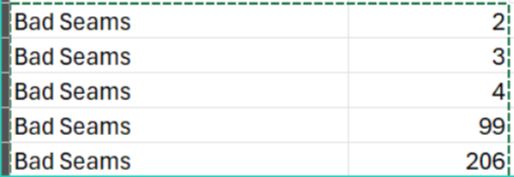
9) Save data after modification and download it :

df\_unique.to\_csv('Defected\_Items\_Unique.csv', index=False)

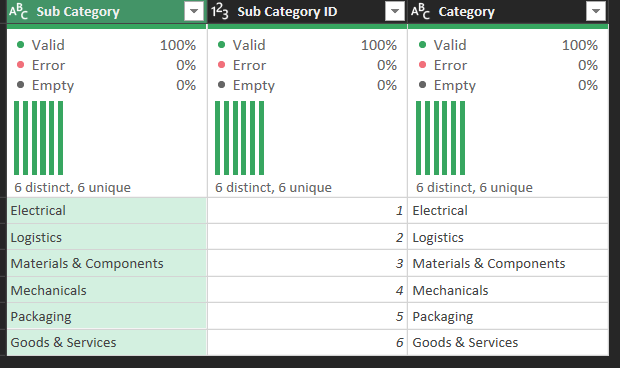
files.download('Defected\_Items\_Unique.csv')

Data Cleaning using Power Query:

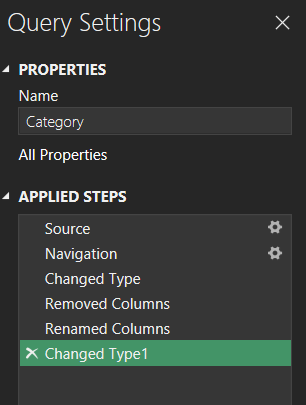
We started by cleaning the data and removing duplicates by replacing it’s original value using Power Query in Excel to ensure accuracy and reliability for further analysis.



Removing ‘sub category’ column from Category sheet cause it’s redundant data.



⬇️



⬇️

