Part1 Data Processing and Cleaning

Prepared by: TAN JUN YUAN (A22EC0107)

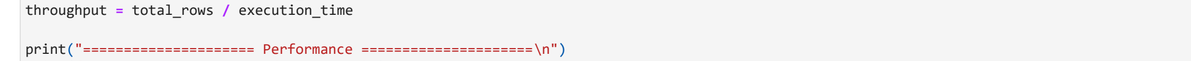
Step 1: Install and Impoet Libraries

In [ ]:
```
!pip install polars
import polars as pl
import pandas as pd
import re
import time
import psutil
import tracemalloc
```

```
Requirement already satisfied: polars in /usr/local/lib/python3.11/dist-packages (1.21.0)
```

Step 2: Upload Excel Files

In [ ]:
```
from google.colab import files
uploaded = files.upload()
```




```
Saving Lazada (Beauty & Skincare).xlsx to Lazada (Beauty & Skincare).xlsx
Saving Lazada (Health & Wellness).xlsx to Lazada (Health & Wellness).xlsx
Saving Lazada (Home & Living).xlsx to Lazada (Home & Living).xlsx
Saving Lazada (Home Appliances).xlsx to Lazada (Home Appliances).xlsx
Saving Lazada (Mother & Baby).xlsx to Lazada (Mother & Baby).xlsx
Saving Lazada (Stationery).xlsx to Lazada (Stationery).xlsx
Saving Lazada (Women's Fashion).xlsx to Lazada (Women's Fashion).xlsx
```

Step 3 : Load Excel Files into PANDAS DataFrames and Check Total Files being Loaded

In [ ]:
```
fList_pandas = [pd.read_excel(file) for file in uploaded.keys()]
print(f"Total File: {len(fList_pandas)}")
```

```
Total File: 7
```

Step 4 : Load and Display Dataset, Checking on Total Numbers of Rows and Columns

In [ ]:
```
%%time
tracemalloc.start()
start_time = time.perf_counter()
total_rows = 0

fList_polars = []
for filename, df in zip(uploaded.keys(), fList_pandas):
    match = re.search(r"\((.*?)\)", filename)
    category = match.group(1) if match else "Unknown"

    pl_df = pl.from_pandas(df).with_columns(pl.lit(category).alias("Category"))
    fList_polars.append(pl_df)

print(f"Total DataFrames: {len(fList_polars)}")

for df in fList_polars:
    total_rows += df.shape[0]
    display(df.head(10))
    print(f"Total rows: {df.shape[0]}")
    print(f"Total columns: {df.shape[1]}\n\n\n")

current, peak = tracemalloc.get_traced_memory()
end_time = time.perf_counter()
tracemalloc.stop()

execution_time = end_time - start_time
throughput = total_rows / execution_time

print("==================== Performance ====================\n")
print(f"Total rows processed: {total_rows}")
print(f"Code Execution time: {execution_time:.4f} seconds")
print(f"Throughput: {throughput:.2f} rows per second")
print(f"Current memory usage: {current / 10**6:.4f} MB")
print(f"Peak memory usage: {peak / 10**6:.4f} MB")

cpu_usage = psutil.cpu_percent(interval=1)
print(f"CPU usage: {cpu_usage}%")

print("=====================================================")

print("\nTotal time for this cell(Including time to display the performance):")
```

```
Total DataFrames: 7
```

shape: (10, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	str	str	str
"❣️SAKA GLOWING FOUNDATION SPF ...	3.99	"Penang"	"55 sold"	"(9)"	"Beauty & Skincare"
"Bio-Essence Bio-Gold 24k Radi...	3.5	"Johor"	"46 sold"	"(10)"	"Beauty & Skincare"
"L'occitane Immortelle Divine F...	1.5	"Selangor"	"13 sold"	"(1)"	"Beauty & Skincare"
"YOUBUY Freckle Cream Effective...	5.45	"China"	"13 sold"	null	"Beauty & Skincare"
"DT37 Jomtam Jolyum Nicotimide ...	6.15	"Melaka"	"175 sold"	"(40)"	"Beauty & Skincare"
"❤️88Home❤️ Jomtam Jolyum Nicot...	6.19	"Melaka"	"29 sold"	"(2)"	"Beauty & Skincare"
"DT37 VEZE Men's Volcanic Mud F...	5.44	"Melaka"	"129 sold"	"(46)"	"Beauty & Skincare"
"NEVEA MEN ROLL ON 500ML"	5.0	"Wp Kuala Lumpur"	"25 sold"	"(4)"	"Beauty & Skincare"
"❤️88Home❤️ VEZE Men's Volcanic...	5.48	"Melaka"	"41 sold"	"(11)"	"Beauty & Skincare"
"DT37 HYMEY'S Facial Cleanser C...	1.53	"Melaka"	"1.3K sold"	"(246)"	"Beauty & Skincare"

```
Total rows: 28325
Total columns: 6
```

shape: (10, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	str	str	str
"HIMALAYA Mentat Tablets 60 (Mi...	15.0	"Wp Kuala Lumpur"	"45 sold"	"(8)"	"Health & Wellness"
"Pil Beauty Original Indonesia ...	5.0	"Selangor"	"56 sold"	"(11)"	"Health & Wellness"
"ELECTRAL FORTE GRANULES (20SX6...	9.9	"Selangor"	"84 sold"	"(22)"	"Health & Wellness"
"[READY STOCK] Spes 1pc Dry Sha...	5.9	"Johor"	"13 sold"	"(3)"	"Health & Wellness"
"CALSONATE Calcium Carbonate Ca...	1.54	"Melaka"	"676 sold"	"(11)"	"Health & Wellness"
"Super Greens, 1350 mg Per Serv...	9.1	"Selangor"	null	null	"Health & Wellness"
"Dyna U Suspension (Peppermint ...	4.0	"Selangor"	"299 sold"	"(10)"	"Health & Wellness"
"🔥Ready Stock🔥Travel Pack 25g M...	10.0	"Selangor"	null	null	"Health & Wellness"
"YSP Homecare Macgel Tablet 10 ...	1.99	"Penang"	"551 sold"	"(21)"	"Health & Wellness"
"Ubat Gastrik/ Gastric/ Sakit P...	2.5	"Wp Kuala Lumpur"	"26 sold"	"(2)"	"Health & Wellness"

```
Total rows: 24382
Total columns: 6
```

shape: (10, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	str	str	str
"80x120/80x160cm Home living ro...	7.71	"Johor"	"4.7K sold"	"(1486)"	"Home & Living"
"80x120/80x160cm Home living ro...	7.79	"Johor"	"860 sold"	"(235)"	"Home & Living"
"Cotton Pillow Sleeping Hilton ...	6.99	"Selangor"	"121 sold"	"(23)"	"Home & Living"
"Sevenland 1pc 70x39cm PE Foam ...	1.8	null	"309.9K sold"	"(4742)"	"Home & Living"
"MISO Foldable Mosquito Net 1.8...	13.34	"Selangor"	"953 sold"	"(180)"	"Home & Living"
"Sleeping Hilton Cotton Pillow ...	12.12	"Johor"	"6 sold"	null	"Home & Living"
"Leego 50CM X 80CM Carpet Mat B...	5.9	"Selangor"	"503 sold"	"(115)"	"Home & Living"
"Exclusive Home & Living Sarung...	4.9	"Selangor"	"106 sold"	"(24)"	"Home & Living"
"3pcs/set Astronaut Decoration ...	3.61	"Selangor"	"206 sold"	"(113)"	"Home & Living"
"Cotton Pillow Sleeping Hilton ...	8.5	"Selangor"	"148 sold"	"(34)"	"Home & Living"

```
Total rows: 13376
Total columns: 6
```

shape: (10, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	str	str	str
"ACTIVEONE Home Appliance House...	31.9	"Penang"	"468 sold"	"(160)"	"Home Appliances"
"Teemo Home Appliance Household...	31.9	"Penang"	"112 sold"	"(39)"	"Home Appliances"
"DESSINI ITALY 250mL USB Rechar...	12.5	"Selangor"	"1.6K sold"	"(611)"	"Home Appliances"
"GTE Home Appliance Household M...	31.9	"Penang"	"229 sold"	"(80)"	"Home Appliances"
"Blender Ready Stock Electric S...	22.88	"Johor"	"2.0K sold"	"(528)"	"Home Appliances"
"2L Electric Meat Grinder Food ...	17.99	"Pahang"	"251 sold"	"(90)"	"Home Appliances"
"Kettle Stainless Steel 2Liter ...	14.5	"Selangor"	"9.0K sold"	"(2286)"	"Home Appliances"
"Electric Jug Kettle 2L Stainle...	17.9	"Selangor"	"4.1K sold"	"(1154)"	"Home Appliances"
"Portable Turbo Electric Fan 10...	19.6	"Selangor"	"7 sold"	"(1)"	"Home Appliances"
"Mafababe Portable Electric Gri...	26.75	"Wp Kuala Lumpur"	"1.0K sold"	"(308)"	"Home Appliances"

```
Total rows: 13674
Total columns: 6
```

shape: (10, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	str	str	str
"RELAXING MOODS FOR MOTHER & BA...	54.9	"Perak"	null	null	"Mother & Baby"
"Cotton Breastfeeding Maternity...	19.9	"Selangor"	"46 sold"	"(10)"	"Mother & Baby"
"Cotton Breastfeeding Nursing C...	8.9	"Selangor"	"802 sold"	"(233)"	"Mother & Baby"
"[Malaysia] Breastfeeding Nursi...	7.5	"Selangor"	"2.9K sold"	"(794)"	"Mother & Baby"
"SummerGlitz 100% Cotton & Cott...	24.9	"Selangor"	"1.1K sold"	"(272)"	"Mother & Baby"
"KOGGY 5 Pairs Maternity Socks ...	11.87	"Johor"	"163 sold"	"(49)"	"Mother & Baby"
"Moo Baby Maternity Socks Stoki...	1.9	"Perak"	"5.2K sold"	"(816)"	"Mother & Baby"
"[PRE-ORDER] Mothers Baby Coolb...	49.33	"Selangor"	null	null	"Mother & Baby"
"Breastfeeding Mum Baby Infant ...	9.99	"Wp Kuala Lumpur"	null	null	"Mother & Baby"
"Einmilk Baby Cotton Nursing Co...	26.9	"Johor"	"25 sold"	"(9)"	"Mother & Baby"

```
Total rows: 2440
Total columns: 6
```

shape: (10, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	str	str	str
"Deli Direct Liquid Gel Pen Qui...	0.7	"Negeri Sembilan"	"764 sold"	"(182)"	"Stationery"
"Colourful Flexible Pencil Soft...	0.39	"Perak"	"508 sold"	"(5)"	"Stationery"
"Stationery 📐 Stationery Set Cart...	0.98	"Perak"	"275 sold"	"(4)"	"Stationery"
"⭐学生卡通中性笔可爱风0.5黑色签字笔⭐New Style ...	0.49	"Perak"	"2.1K sold"	"(148)"	"Stationery"
"*Original* M&G R3/R5 0.5 0.7 G...	0.75	"Selangor"	"64.8K sold"	"(4015)"	"Stationery"
"12PCS/Set Basics Premium Multi...	0.5	"Selangor"	"55.4K sold"	"(12408)"	"Stationery"
"（No need to sharpen pencils）Fr...	0.54	"Selangor"	"8.5K sold"	"(702)"	"Stationery"
"⭐学生卡通本创意文具高颜值笔记本子学习用品记事本⭐Noteb...	0.29	"Perak"	"3.5K sold"	"(40)"	"Stationery"
"Classic Gel Pen 1008 Black Blu...	0.39	"Perak"	"16.2K sold"	"(492)"	"Stationery"
"Sanrio Eraser Stationery Non-D...	0.96	"Selangor"	"1.1K sold"	"(183)"	"Stationery"

```
Total rows: 23195
Total columns: 6
```

shape: (10, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	str	str	str
"Upsee Women's Fashion Heat Res…	17.13	"China"	"753 sold"	"(188)"	"Women's Fashion"
"Summer women's fashionable hig…	13.0	"China"	"54 sold"	"(14)"	"Women's Fashion"
"Summer New Sports Suit Women's…	22.75	"China"	"286 sold"	"(62)"	"Women's Fashion"
"Daring Backless V-Neck Dress S…	19.72	null	"270 sold"	"(37)"	"Women's Fashion"
"Chic Lace Birthday Dress Chris…	12.18	null	"486 sold"	"(35)"	"Women's Fashion"
"LOMOGI Women's Fashion Dress +…	15.23	"China"	"1.6K sold"	"(455)"	"Women's Fashion"
"Summer 2024 Women's Knitted Lo…	22.41	null	"322 sold"	"(26)"	"Women's Fashion"
"Hot Princess Dress Set Sexy Sl…	17.39	null	"92 sold"	"(6)"	"Women's Fashion"
"LOMOGI Women's Fashion Dress +…	14.17	null	"122 sold"	"(42)"	"Women's Fashion"
"Adult Latin Dance Skirt Square…	20.55	null	"350 sold"	"(58)"	"Women's Fashion"

```
Total rows: 9698
Total columns: 6


```

```
==================== Performance ====================

Total rows processed: 115090
Code Execution time: 0.6578 seconds
Throughput: 174953.89 rows per second
Current memory usage: 2.7757 MB
Peak memory usage: 2.8360 MB
CPU usage: 57.8%
=====================================================

Total time for this cell(Including time to display the performance):
CPU times: user 337 ms, sys: 42.8 ms, total: 380 ms
Wall time: 1.67 s
```

Step 5 : Combine All DataFrames into One (Data Integration), Checking on Total Numbers of Rows and Columns

In [ ]:

```
%%time

tracemalloc.start()
start_time = time.perf_counter()

df_combined = pl.concat(fList_polars, how="vertical", rechunk=True)

df_combined.write_csv("polars_combined_dataset.csv")
files.download("polars_combined_dataset.csv")

display(df_combined.head(10))
total_rows = df_combined.shape[0]
print(f"Total rows: {df_combined.shape[0]}")
print(f"Total columns: {df_combined.shape[1]}\n\n\n")

current, peak = tracemalloc.get_traced_memory()
end_time = time.perf_counter()
tracemalloc.stop()

execution_time = end_time - start_time
throughput = total_rows / execution_time

print("==================== Performance ====================\n")
print(f"Total rows processed: {total_rows}")
print(f"Code Execution time: {execution_time:.4f} seconds")
print(f"Throughput: {throughput:.2f} rows per second")
print(f"Current memory usage: {current / 10**6:.4f} MB")
print(f"Peak memory usage: {peak / 10**6:.4f} MB")

cpu_usage = psutil.cpu_percent(interval=1)
print(f"CPU usage: {cpu_usage}%")

print("=====================================================")

print("\nTotal time for this cell(Including time to display the performance):")
```

shape: (10, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	str	str	str
"❣️SAKA GLOWING FOUNDATION SPF ...	3.99	"Penang"	"55 sold"	"(9)"	"Beauty & Skincare"
"Bio-Essence Bio-Gold 24k Radi...	3.5	"Johor"	"46 sold"	"(10)"	"Beauty & Skincare"
"L'occitane Immortelle Divine F...	1.5	"Selangor"	"13 sold"	"(1)"	"Beauty & Skincare"
"YOUBUY Freckle Cream Effective...	5.45	"China"	"13 sold"	null	"Beauty & Skincare"
"DT37 Jomtam Jolyum Nicotimide ...	6.15	"Melaka"	"175 sold"	"(40)"	"Beauty & Skincare"
"❤️88Home❤️ Jomtam Jolyum Nicot...	6.19	"Melaka"	"29 sold"	"(2)"	"Beauty & Skincare"
"DT37 VEZE Men's Volcanic Mud F...	5.44	"Melaka"	"129 sold"	"(46)"	"Beauty & Skincare"
"NEVEA MEN ROLL ON 500ML"	5.0	"Wp Kuala Lumpur"	"25 sold"	"(4)"	"Beauty & Skincare"
"❤️88Home❤️ VEZE Men's Volcanic...	5.48	"Melaka"	"41 sold"	"(11)"	"Beauty & Skincare"
"DT37 HYMEY'S Facial Cleanser C...	1.53	"Melaka"	"1.3K sold"	"(246)"	"Beauty & Skincare"

```
Total rows: 115090
Total columns: 6


==================== Performance ====================

Total rows processed: 115090
Code Execution time: 0.1654 seconds
Throughput: 695962.63 rows per second
Current memory usage: 0.0284 MB
Peak memory usage: 0.0989 MB
CPU usage: 4.0%
=====================================================

Total time for this cell(Including time to display the performance):
CPU times: user 95.5 ms, sys: 47.3 ms, total: 143 ms
Wall time: 1.17 s
```

Step 6 : Standardizing Field with Object/String Data Type into Uppercase

In [ ]:

```
%%time

tracemalloc.start()
start_time = time.perf_counter()
total_rows = df_combined.shape[0]

df_combined = df_combined.with_columns([
    pl.col(col).str.to_uppercase().alias(col)
    for col, dtype in zip(df_combined.columns, df_combined.dtypes)
    if dtype == pl.Utf8
])

display(df_combined.head(10))
print(f"Total rows: {df_combined.shape[0]}")
print(f"Total columns: {df_combined.shape[1]}\n\n\n")

current, peak = tracemalloc.get_traced_memory()
end_time = time.perf_counter()
tracemalloc.stop()

execution_time = end_time - start_time
throughput = total_rows / execution_time

print("==================== Performance ====================\n")
print(f"Total rows processed: {total_rows}")
print(f"Code Execution time: {execution_time:.4f} seconds")
print(f"Throughput: {throughput:.2f} rows per second")
print(f"Current memory usage: {current / 10**6:.4f} MB")
print(f"Peak memory usage: {peak / 10**6:.4f} MB")

cpu_usage = psutil.cpu_percent(interval=1)
print(f"CPU usage: {cpu_usage}%")

print("=====================================================")

print("\nTotal time for this cell(Including time to display the performance):")
```

shape: (10, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	str	str	str
"❣️ SAKA GLOWING FOUNDATION SPF ...	3.99	"PENANG"	"55 SOLD"	"(9)"	"BEAUTY & SKINCARE"
"BIO-ESSENCE BIO-GOLD 24K RADI...	3.5	"JOHOR"	"46 SOLD"	"(10)"	"BEAUTY & SKINCARE"
"L'OCCITANE IMMORTELLE DIVINE F...	1.5	"SELANGOR"	"13 SOLD"	"(1)"	"BEAUTY & SKINCARE"
"YOUBUY FRECKLE CREAM EFFECTIVE...	5.45	"CHINA"	"13 SOLD"	null	"BEAUTY & SKINCARE"
"DT37 JOMTAM JOLYUM NICOTIMIDE ...	6.15	"MELAKA"	"175 SOLD"	"(40)"	"BEAUTY & SKINCARE"
"❤️88HOME❤️ JOMTAM JOLYUM NICOT...	6.19	"MELAKA"	"29 SOLD"	"(2)"	"BEAUTY & SKINCARE"
"DT37 VEZE MEN'S VOLCANIC MUD F...	5.44	"MELAKA"	"129 SOLD"	"(46)"	"BEAUTY & SKINCARE"
"NEVEA MEN ROLL ON 500ML"	5.0	"WP KUALA LUMPUR"	"25 SOLD"	"(4)"	"BEAUTY & SKINCARE"
"❤️88HOME❤️ VEZE MEN'S VOLCANIC...	5.48	"MELAKA"	"41 SOLD"	"(11)"	"BEAUTY & SKINCARE"
"DT37 HYMEY'S FACIAL CLEANSER C...	1.53	"MELAKA"	"1.3K SOLD"	"(246)"	"BEAUTY & SKINCARE"

```
Total rows: 115090
Total columns: 6


==================== Performance ====================

Total rows processed: 115090
Code Execution time: 0.0763 seconds
Throughput: 1507622.39 rows per second
Current memory usage: 0.0114 MB
Peak memory usage: 0.0413 MB
CPU usage: 4.0%
=====================================================

Total time for this cell(Including time to display the performance):
CPU times: user 91.7 ms, sys: 21.8 ms, total: 113 ms
Wall time: 1.08 s
```

Step 7 : Converting 'Total Reviews' into Integer Data Type

In [ ]:

```
%%time

tracemalloc.start()
start_time = time.perf_counter()
total_rows = df_combined.shape[0]

df_combined = df_combined.with_columns([
    pl.col("Total Reviews")
      .str.replace_all(r"[^\d]", "")
      .cast(pl.Int64)
      .alias("Total Reviews")
])

display(df_combined.head(10))
print(f"Total rows: {df_combined.shape[0]}")
print(f"Total columns: {df_combined.shape[1]}\n\n\n")

current, peak = tracemalloc.get_traced_memory()
end_time = time.perf_counter()
tracemalloc.stop()

execution_time = end_time - start_time
throughput = total_rows / execution_time

print("==================== Performance ====================\n")
print(f"Total rows processed: {total_rows}")
print(f"Code Execution time: {execution_time:.4f} seconds")
print(f"Throughput: {throughput:.2f} rows per second")
print(f"Current memory usage: {current / 10**6:.4f} MB")
print(f"Peak memory usage: {peak / 10**6:.4f} MB")

cpu_usage = psutil.cpu_percent(interval=1)
print(f"CPU usage: {cpu_usage}%")

print("=====================================================")

print("\nTotal time for this cell(Including time to display the performance):")
```

shape: (10, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	str	i64	str
"❣️ SAKA GLOWING FOUNDATION SPF ...	3.99	"PENANG"	"55 SOLD"	9	"BEAUTY & SKINCARE"
"BIO-ESSENCE BIO-GOLD 24K RADI...	3.5	"JOHOR"	"46 SOLD"	10	"BEAUTY & SKINCARE"
"L'OCCITANE IMMORTELLE DIVINE F...	1.5	"SELANGOR"	"13 SOLD"	1	"BEAUTY & SKINCARE"
"YOUBUY FRECKLE CREAM EFFECTIVE...	5.45	"CHINA"	"13 SOLD"	null	"BEAUTY & SKINCARE"
"DT37 JOMTAM JOLYUM NICOTIMIDE ...	6.15	"MELAKA"	"175 SOLD"	40	"BEAUTY & SKINCARE"
"❤️88HOME❤️ JOMTAM JOLYUM NICOT...	6.19	"MELAKA"	"29 SOLD"	2	"BEAUTY & SKINCARE"
"DT37 VEZE MEN'S VOLCANIC MUD F...	5.44	"MELAKA"	"129 SOLD"	46	"BEAUTY & SKINCARE"
"NEVEA MEN ROLL ON 500ML"	5.0	"WP KUALA LUMPUR"	"25 SOLD"	4	"BEAUTY & SKINCARE"
"❤️88HOME❤️ VEZE MEN'S VOLCANIC...	5.48	"MELAKA"	"41 SOLD"	11	"BEAUTY & SKINCARE"
"DT37 HYMEY'S FACIAL CLEANSER C...	1.53	"MELAKA"	"1.3K SOLD"	246	"BEAUTY & SKINCARE"

```
Total rows: 115090
Total columns: 6


==================== Performance ====================

Total rows processed: 115090
Code Execution time: 0.0359 seconds
Throughput: 3204506.72 rows per second
Current memory usage: 0.0106 MB
Peak memory usage: 0.0398 MB
CPU usage: 4.5%
=====================================================

Total time for this cell(Including time to display the performance):
CPU times: user 39.7 ms, sys: 3.08 ms, total: 42.8 ms
Wall time: 1.04 s
```

Step 8 : Converting 'Quatity Sold' into Integer Data Type

In [ ]:

```
%%time

tracemalloc.start()
start_time = time.perf_counter()
total_rows = df_combined.shape[0]

df_combined = df_combined.with_columns([
    pl.col("Quantity Sold")
      .str.replace_all("SOLD", "")
      .str.strip_chars()
      .alias("Quantity Sold")
])

display(df_combined.head(10))
print(f"Total rows: {df_combined.shape[0]}")
print(f"Total columns: {df_combined.shape[1]}")
print("\n")

has_alpha = df_combined.filter(
    pl.col("Quantity Sold").str.contains(r"[A-Z]")
)

text = " ".join(has_alpha["Quantity Sold"].to_list())
unique_letters = sorted(set(re.findall(r"[A-Z]", text)))

print("Unique alphabetic characters found:", unique_letters)

df_combined = df_combined.with_columns([
    pl.when(pl.col("Quantity Sold").str.contains("K"))
      .then(
          pl.col("Quantity Sold")
            .str.replace_all("K", "")
            .str.replace_all(r"[^\d\.]", "")
            .cast(pl.Float64) * 1000
      )
      .otherwise(
          pl.col("Quantity Sold")
            .str.replace_all(r"[^\d]", "")
            .cast(pl.Float64)
      )
      .cast(pl.Int64)
      .alias("Quantity Sold")
])
```

```
display(df_combined.head(10))
print(f"Total rows: {df_combined.shape[0]}")
print(f"Total columns: {df_combined.shape[1]}")
print("\n\n\n")

current, peak = tracemalloc.get_traced_memory()
end_time = time.perf_counter()
tracemalloc.stop()

execution_time = end_time - start_time
throughput = total_rows / execution_time

print("==================== Performance ====================\n")
print(f"Total rows processed: {total_rows}")
print(f"Code Execution time: {execution_time:.4f} seconds")
print(f"Throughput: {throughput:.2f} rows per second")
print(f"Current memory usage: {current / 10**6:.4f} MB")
print(f"Peak memory usage: {peak / 10**6:.4f} MB")

cpu_usage = psutil.cpu_percent(interval=1)
print(f"CPU usage: {cpu_usage}%")

print("=====================================================")

print("\nTotal time for this cell(Including time to display the performance):")
```

shape: (10, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	str	i64	str
"❣️ SAKA GLOWING FOUNDATION SPF ...	3.99	"PENANG"	"55"	9	"BEAUTY & SKINCARE"
"BIO-ESSENCE BIO-GOLD 24K RADI...	3.5	"JOHOR"	"46"	10	"BEAUTY & SKINCARE"
"L'OCCITANE IMMORTELLE DIVINE F...	1.5	"SELANGOR"	"13"	1	"BEAUTY & SKINCARE"
"YOUBUY FRECKLE CREAM EFFECTIVE...	5.45	"CHINA"	"13"	null	"BEAUTY & SKINCARE"
"DT37 JOMTAM JOLYUM NICOTIMIDE ...	6.15	"MELAKA"	"175"	40	"BEAUTY & SKINCARE"
"❤️88HOME❤️ JOMTAM JOLYUM NICOT...	6.19	"MELAKA"	"29"	2	"BEAUTY & SKINCARE"
"DT37 VEZE MEN'S VOLCANIC MUD F...	5.44	"MELAKA"	"129"	46	"BEAUTY & SKINCARE"
"NEVEA MEN ROLL ON 500ML"	5.0	"WP KUALA LUMPUR"	"25"	4	"BEAUTY & SKINCARE"
"❤️88HOME❤️ VEZE MEN'S VOLCANIC...	5.48	"MELAKA"	"41"	11	"BEAUTY & SKINCARE"
"DT37 HYMEY'S FACIAL CLEANSER C...	1.53	"MELAKA"	"1.3K"	246	"BEAUTY & SKINCARE"

```
Total rows: 115090
Total columns: 6


 Unique alphabetic characters found: ['K']
```

shape: (10, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	i64	i64	str
"❣️ SAKA GLOWING FOUNDATION SPF ...	3.99	"PENANG"	55	9	"BEAUTY & SKINCARE"
"BIO-ESSENCE BIO-GOLD 24K RADI...	3.5	"JOHOR"	46	10	"BEAUTY & SKINCARE"
"L'OCCITANE IMMORTELLE DIVINE F...	1.5	"SELANGOR"	13	1	"BEAUTY & SKINCARE"
"YOUBUY FRECKLE CREAM EFFECTIVE...	5.45	"CHINA"	13	null	"BEAUTY & SKINCARE"
"DT37 JOMTAM JOLYUM NICOTIMIDE ...	6.15	"MELAKA"	175	40	"BEAUTY & SKINCARE"
"❤️88HOME❤️ JOMTAM JOLYUM NICOT...	6.19	"MELAKA"	29	2	"BEAUTY & SKINCARE"
"DT37 VEZE MEN'S VOLCANIC MUD F...	5.44	"MELAKA"	129	46	"BEAUTY & SKINCARE"
"NEVEA MEN ROLL ON 500ML"	5.0	"WP KUALA LUMPUR"	25	4	"BEAUTY & SKINCARE"
"❤️88HOME❤️ VEZE MEN'S VOLCANIC...	5.48	"MELAKA"	41	11	"BEAUTY & SKINCARE"
"DT37 HYMEY'S FACIAL CLEANSER C...	1.53	"MELAKA"	1300	246	"BEAUTY & SKINCARE"

```
Total rows: 115090
Total columns: 6



==================== Performance ====================

Total rows processed: 115090
Code Execution time: 0.1205 seconds
Throughput: 955122.88 rows per second
Current memory usage: 0.0490 MB
Peak memory usage: 0.3903 MB
CPU usage: 4.0%
=====================================================

Total time for this cell(Including time to display the performance):
CPU times: user 83.6 ms, sys: 7.92 ms, total: 91.5 ms
Wall time: 1.12 s
```

Step 9 : Checking and Handling Missing Values by Replacing 0 for Numeric Fields and N/A for String/Object Fields

In [ ]:

```
%%time
tracemalloc.start()
start_time = time.perf_counter()
total_rows = df_combined.shape[0]

print("Initial Dataset:")
display(df_combined.head(10))

print("Before Handle Missing Values")
missing_values = df_combined.select([pl.col(col).is_null().sum().alias(f"{col}_missing") for col in df_combined.columns])
print("Number of Missing Values for Each Column:")
display(missing_values)

df_combined = df_combined.with_columns([
    (
        pl.col(col).fill_null("N/A") if dtype == pl.Utf8
        else pl.col(col).fill_null(0) if dtype in (pl.Int8, pl.Int16, pl.Int32, pl.Int64,
                                                   pl.UInt8, pl.UInt16, pl.UInt32, pl.UInt64,
                                                   pl.Float32, pl.Float64)
        else pl.col(col)
    ).alias(col)
    for col, dtype in zip(df_combined.columns, df_combined.dtypes)
])


print("\nAfter Handle Missing Values")
missing_values = df_combined.select([pl.col(col).is_null().sum().alias(f"{col}_missing") for col in df_combined.columns])
print("Number of Missing Values for Each Column:")
display(missing_values)

print("\nFinalised Dataset:")
display(df_combined.head(10))
print(f"Total rows: {df_combined.shape[0]}")
print(f"Total columns: {df_combined.shape[1]}\n\n\n")


current, peak = tracemalloc.get_traced_memory()
end_time = time.perf_counter()
tracemalloc.stop()

execution_time = end_time - start_time
throughput = total_rows / execution_time

print("==================== Performance ====================\n")
print(f"Total rows processed: {total_rows}")
print(f"Code Execution time: {execution_time:.4f} seconds")
print(f"Throughput: {throughput:.2f} rows per second")
print(f"Current memory usage: {current / 10**6:.4f} MB")
print(f"Peak memory usage: {peak / 10**6:.4f} MB")

cpu_usage = psutil.cpu_percent(interval=1)
print(f"CPU usage: {cpu_usage}%")

print("=====================================================")

print("\nTotal time for this cell(Including time to display the performance):")
```

```
Initial Dataset:
```

shape: (10, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	i64	i64	str
"❣️ SAKA GLOWING FOUNDATION SPF ...	3.99	"PENANG"	55	9	"BEAUTY & SKINCARE"
"BIO-ESSENCE BIO-GOLD 24K RADI...	3.5	"JOHOR"	46	10	"BEAUTY & SKINCARE"
"L'OCCITANE IMMORTELLE DIVINE F...	1.5	"SELANGOR"	13	1	"BEAUTY & SKINCARE"
"YOUBUY FRECKLE CREAM EFFECTIVE...	5.45	"CHINA"	13	null	"BEAUTY & SKINCARE"
"DT37 JOMTAM JOLYUM NICOTIMIDE ...	6.15	"MELAKA"	175	40	"BEAUTY & SKINCARE"
"❤️88HOME❤️ JOMTAM JOLYUM NICOT...	6.19	"MELAKA"	29	2	"BEAUTY & SKINCARE"
"DT37 VEZE MEN'S VOLCANIC MUD F...	5.44	"MELAKA"	129	46	"BEAUTY & SKINCARE"
"NEVEA MEN ROLL ON 500ML"	5.0	"WP KUALA LUMPUR"	25	4	"BEAUTY & SKINCARE"
"❤️88HOME❤️ VEZE MEN'S VOLCANIC...	5.48	"MELAKA"	41	11	"BEAUTY & SKINCARE"
"DT37 HYMEY'S FACIAL CLEANSER C...	1.53	"MELAKA"	1300	246	"BEAUTY & SKINCARE"

```
Before Handle Missing Values
Number of Missing Values for Each Column:
```

shape: (1, 6)

Product Name_missing	Price_missing	Location_missing	Quantity Sold_missing	Total Reviews_missing	Category_missing
u32	u32	u32	u32	u32	u32
2	0	13517	45566	50958	0

```
After Handle Missing Values
Number of Missing Values for Each Column:
```

shape: (1, 6)

Product Name_missing	Price_missing	Location_missing	Quantity Sold_missing	Total Reviews_missing	Category_missing
u32	u32	u32	u32	u32	u32
0	0	0	0	0	0

```
Finalised Dataset:
```

shape: (10, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	i64	i64	str
"❣️ SAKA GLOWING FOUNDATION SPF ...	3.99	"PENANG"	55	9	"BEAUTY & SKINCARE"
"BIO-ESSENCE BIO-GOLD 24K RADI...	3.5	"JOHOR"	46	10	"BEAUTY & SKINCARE"
"L'OCCITANE IMMORTELLE DIVINE F...	1.5	"SELANGOR"	13	1	"BEAUTY & SKINCARE"
"YOUBUY FRECKLE CREAM EFFECTIVE...	5.45	"CHINA"	13	0	"BEAUTY & SKINCARE"
"DT37 JOMTAM JOLYUM NICOTIMIDE ...	6.15	"MELAKA"	175	40	"BEAUTY & SKINCARE"
"❤️88HOME❤️ JOMTAM JOLYUM NICOT...	6.19	"MELAKA"	29	2	"BEAUTY & SKINCARE"
"DT37 VEZE MEN'S VOLCANIC MUD F...	5.44	"MELAKA"	129	46	"BEAUTY & SKINCARE"
"NEVEA MEN ROLL ON 500ML"	5.0	"WP KUALA LUMPUR"	25	4	"BEAUTY & SKINCARE"
"❤️88HOME❤️ VEZE MEN'S VOLCANIC...	5.48	"MELAKA"	41	11	"BEAUTY & SKINCARE"
"DT37 HYMEY'S FACIAL CLEANSER C...	1.53	"MELAKA"	1300	246	"BEAUTY & SKINCARE"

```
Total rows: 115090
Total columns: 6


==================== Performance ====================

Total rows processed: 115090
Code Execution time: 0.0589 seconds
Throughput: 1953944.00 rows per second
Current memory usage: 0.0505 MB
Peak memory usage: 0.1000 MB
CPU usage: 4.0%
=====================================================

Total time for this cell(Including time to display the performance):
CPU times: user 51.5 ms, sys: 11.8 ms, total: 63.3 ms
Wall time: 1.06 s
```

Step 10 : Checking and Handling Duplicate Rows and Displaying in a View that Arranges Duplicate Rows Together

In [ ]:
```
%%time

tracemalloc.start()
start_time = time.perf_counter()
total_rows = df_combined.shape[0]

duplicate_rows = df_combined.filter(df_combined.is_duplicated()).sort(df_combined.columns)

print("Before Handling duplicate")
print(f"Number of duplicate rows: {duplicate_rows.shape[0]}")
display(duplicate_rows)
print(f"Total rows: {duplicate_rows.shape[0]}")
print(f"Total columns: {duplicate_rows.shape[1]}")

df_cleaned = df_combined.unique()

print("\nAfter Handling duplicate")
duplicate_rows = df_cleaned.filter(df_cleaned.is_duplicated()).sort(df_cleaned.columns)
print(f"Number of duplicate rows: {duplicate_rows.shape[0]}")
display(duplicate_rows)
print(f"Total rows: {duplicate_rows.shape[0]}")
print(f"Total columns: {duplicate_rows.shape[1]}")

print("\nFinalised Dataset:")
display(df_cleaned.head(10))
print(f"Total rows: {df_cleaned.shape[0]}")
print(f"Total columns: {df_cleaned.shape[1]}\n\n\n")

current, peak = tracemalloc.get_traced_memory()
end_time = time.perf_counter()
tracemalloc.stop()

execution_time = end_time - start_time
throughput = total_rows / execution_time

print("==================== Performance ====================\n")
print(f"Total rows processed: {total_rows}")
print(f"Code Execution time: {execution_time:.4f} seconds")
print(f"Throughput: {throughput:.2f} rows per second")
print(f"Current memory usage: {current / 10**6:.4f} MB")
print(f"Peak memory usage: {peak / 10**6:.4f} MB")

cpu_usage = psutil.cpu_percent(interval=1)
print(f"CPU usage: {cpu_usage}%")

print("=====================================================")

print("\nTotal time for this cell(Including time to display the performance):")
```

```
Before Handling duplicate
Number of duplicate rows: 2854
```

shape: (2_854, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	i64	i64	str
"#ARYAN&RAIHAN OOTHING ALOE VER...	12.0	"WP KUALA LUMPUR"	12	4	"BEAUTY & SKINCARE"
"#ARYAN&RAIHAN OOTHING ALOE VER...	12.0	"WP KUALA LUMPUR"	12	4	"BEAUTY & SKINCARE"
"(ROHTO) HADA-LABO GOKUJUN PR...	50.19	"JAPAN"	0	1	"BEAUTY & SKINCARE"
"(ROHTO) HADA-LABO GOKUJUN PR...	50.19	"JAPAN"	0	1	"BEAUTY & SKINCARE"
"(1 PCS) 70X77 3D WALLPAPER BRI...	1.98	"PERAK"	3400	93	"HOME & LIVING"
...	...	...	...	...	...
"🔥READY STOCK🔥超便宜超便宜PROMOTION 1...	47.0	"WP KUALA LUMPUR"	8	5	"BEAUTY & SKINCARE"
"🔥STOK SEDIA ADA🔥 TANAMERA BLAC...	22.0	"PENANG"	0	0	"BEAUTY & SKINCARE"
"🔥STOK SEDIA ADA🔥 TANAMERA BLAC...	22.0	"PENANG"	0	0	"BEAUTY & SKINCARE"
"🥥 COCONUT OIL NATURAL LIP BALM...	12.0	"SELANGOR"	0	0	"BEAUTY & SKINCARE"
"🥥 COCONUT OIL NATURAL LIP BALM...	12.0	"SELANGOR"	0	0	"BEAUTY & SKINCARE"

```
Total rows: 2854
Total columns: 6

After Handling duplicate
Number of duplicate rows: 0
```

shape: (0, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	i64	i64	str

```
Total rows: 0
Total columns: 6

Finalised Dataset:
```
shape: (10, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	i64	i64	str
"NEW PROMO PERSPIREX COMFORT EX...	45.86	"JOHOR"	0	0	"BEAUTY & SKINCARE"
"NEW YEAR CARTOON CAT LUCKY CAT...	21.7	"N/A"	83	10	"HOME & LIVING"
"ESW CAPSULE WHITENING / ESW BO...	39.0	"PERAK"	6	3	"HEALTH & WELLNESS"
"GLUCO DR. AUTO TEST STRIPS 25'...	85.0	"WP KUALA LUMPUR"	0	2	"HEALTH & WELLNESS"
"RETRACTABLE 0.5MM RED BLUE BLA...	4.42	"CHINA"	314	0	"STATIONERY"
"ARTLINE 500A WHITEBOARD MARKER...	6.85	"NEGERI SEMBILAN"	0	0	"STATIONERY"
"[CHRISTMAS & NEW YEAR GIFTS] C...	89.0	"CHINA"	25	10	"STATIONERY"
"FRENCH ELEGANT SEXY LONG DRESS...	45.01	"N/A"	61	18	"WOMEN'S FASHION"
"LUOFAN BEAUTY VASELINE PELEMBA...	51.0	"CHINA"	0	0	"BEAUTY & SKINCARE"
"【NUTRIENT BOOST】MILK THISTLE S...	16.43	"SELANGOR"	60	19	"HEALTH & WELLNESS"

```
Total rows: 113596
Total columns: 6


==================== Performance ====================

Total rows processed: 115090
Code Execution time: 0.1976 seconds
Throughput: 582459.45 rows per second
Current memory usage: 0.0520 MB
Peak memory usage: 0.0750 MB
CPU usage: 4.5%
======================================================

Total time for this cell(Including time to display the performance):
CPU times: user 232 ms, sys: 86.5 ms, total: 319 ms
Wall time: 1.2 s
```

Step 11 : Exporting a Cleaned Excel Data File for Data Optimization

In [ ]:
```python
%%time
tracemalloc.start()
start_time = time.perf_counter()
total_rows = df_cleaned.shape[0]

df_cleaned.write_csv("polars_cleaned_dataset.csv")
files.download("polars_cleaned_dataset.csv")

current, peak = tracemalloc.get_traced_memory()
end_time = time.perf_counter()
tracemalloc.stop()

execution_time = end_time - start_time
throughput = total_rows / execution_time

print("==================== Performance ====================\n")
print(f"Total rows processed: {total_rows}")
print(f"Code Execution time: {execution_time:.4f} seconds")
print(f"Throughput: {throughput:.2f} rows per second")
print(f"Current memory usage: {current / 10**6:.4f} MB")
print(f"Peak memory usage: {peak / 10**6:.4f} MB")

cpu_usage = psutil.cpu_percent(interval=1)
print(f"CPU usage: {cpu_usage}%")

print("======================================================")

print("\nTotal time for this cell(Including time to display the performance):")
```

```
==================== Performance ====================

Total rows processed: 113596
Code Execution time: 0.0959 seconds
Throughput: 1184103.69 rows per second
Current memory usage: 0.0081 MB
Peak memory usage: 0.0143 MB
CPU usage: 4.0%
=====================================================

Total time for this cell(Including time to display the performance):
CPU times: user 112 ms, sys: 33.7 ms, total: 145 ms
Wall time: 1.1 s
```

End of Part 1 Data Processing and Cleaning