DATA ANALYST

Intenship Task 14

DESCRIPTION

The task involved building a mini ETL (Extract, Transform, Load) pipeline using Python and Pandas. A cleaned Superstore dataset was extracted, transformed through data cleaning and feature engineering, and loaded into structured CSV outputs and a SQLite database to understand practical ETL workflows.

PREPARED BY

Reema Safrin M
(10-02-2026)

MY WORK

I implemented an end-to-end ETL mini pipeline using Python in Google Colab with the Superstore dataset. The process began by extracting the dataset and organizing a structured folder system for raw, processed, and output data. I performed data cleaning by handling missing values, removing duplicates, standardizing column names, and converting date formats. Additional derived columns such as profit margin and delivery days were created to enhance analysis. The transformed data was then split into customers, orders, and products tables and exported as CSV files. Finally, I loaded the outputs into a SQLite database and validated record counts to ensure data accuracy and consistency.

DATASET USED

superstore dataset

FINAL DATASET

processed_superstore

MY WORK

In [1]:
```
import pandas as pd
import os
import sqlite3
```

In [3]:
```
os.listdir()
```

Out[3]:
```
['.config', 'sample_superstore_clean(superstore dataset).csv', 'sample_data']
```

In [4]:
```
df = pd.read_csv("sample_superstore_clean(superstore dataset).csv")
df.head()
```

Out[4]:

	Row_ID	Order_ID	Order_Date	Ship_Date	Ship_Mode	Customer_ID	Customer_Name
0	1	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute
1	2	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute
2	3	CA-2016-138688	6/12/2016	6/16/2016	Second Class	DV-13045	Darrin Van Huff
3	4	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell
4	5	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell

5 rows × 21 columns

In [5]:
```
os.makedirs("raw", exist_ok=True)
os.makedirs("processed", exist_ok=True)
os.makedirs("output", exist_ok=True)
```

In [6]:
```
df.to_csv("raw/superstore_raw.csv", index=False)
```

In [7]:
```
df.isnull().sum()
```

Out[7]:

	0
Row_ID	0
Order_ID	0
Order_Date	0
Ship_Date	0
Ship_Mode	0
Customer_ID	0
Customer_Name	0
Segment	0
Country	0
City	0
State	0
Postal_Code	0
Region	0
Product_ID	0
Category	0
Sub_Category	0
Product_Name	0
Sales	0
Quantity	0
Discount	0
Profit	0

dtype: int64

In [8]:
```
df = df.dropna()
```

In [9]:
```
df = df.drop_duplicates()
```

In [10]:
```
df.columns = df.columns.str.lower().str.replace(" ", "_")
```

In [11]:
```
df['order_date'] = pd.to_datetime(df['order_date'])
df['ship_date'] = pd.to_datetime(df['ship_date'])
```

In [12]:
```
df['profit_margin'] = df['profit'] / df['sales']
df['delivery_days'] = (df['ship_date'] - df['order_date']).dt.days
```

In [13]:
```
df['high_value_order'] = df['sales'].apply(lambda x: 'Yes' if x > 500 else 'No')
```

In [14]:
```
customers = df[['customer_id', 'customer_name', 'segment', 'country', 'region']].dr
```

In [15]:
```
orders = df[['order_id', 'order_date', 'ship_date', 'ship_mode', 'customer_id', 'sa
```

In [16]:
```
products = df[['product_id', 'category', 'sub_category', 'product_name']]
```

In [17]:
```
customers.to_csv("output/customers.csv", index=False)
orders.to_csv("output/orders.csv", index=False)
products.to_csv("output/products.csv", index=False)
```

In [18]:
```
conn = sqlite3.connect("database.sqlite")

customers.to_sql("customers", conn, if_exists="replace", index=False)
orders.to_sql("orders", conn, if_exists="replace", index=False)
products.to_sql("products", conn, if_exists="replace", index=False)

conn.close()
```

In [19]:
```
print("Total records:", len(df))
print("Customers:", len(customers))
print("Orders:", len(orders))
print("Products:", len(products))
```

```
Total records: 9994
Customers: 2501
Orders: 9994
Products: 9994
```

In [20]:
```
df.to_csv("processed/processed_superstore.csv", index=False)
```