Retail Sales EDA Project

```
#IMporting libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
#loading the data
df = pd.read_csv('/content/retail_sales_dataset.csv')
```

1. Load and dataset summary

```
#Dataset contains 1000rows and 9columns
df = pd.read_csv('/content/retail_sales_dataset.csv')
```

```
# total columns(9)
df.head(9) #first 5 rows
```

```
   Transaction ID        Date Customer ID  Gender  Age Product Category  Quantity  Price per Unit  Total Amount
0               1  2023-11-24     CUST001    Male   34           Beauty         3              50           150
1               2  2023-02-27     CUST002  Female   26         Clothing         2             500          1000
2               3  2023-01-13     CUST003    Male   50      Electronics         1              30            30
3               4  2023-05-21     CUST004    Male   37         Clothing         1             500           500
4               5  2023-05-06     CUST005    Male   30           Beauty         2              50           100
5               6  2023-04-25     CUST006  Female   45           Beauty         1              30            30
6               7  2023-03-13     CUST007    Male   46         Clothing         2              25            50
7               8  2023-02-22     CUST008    Male   30      Electronics         4              25           100
8               9  2023-12-13     CUST009    Male   63      Electronics         2             300           600
```

Next steps: Generate code with df New interactive sheet

```
df
```

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount
0	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
3	4	2023-05-21	CUST004	Male	37	Clothing	1	500	500
4	5	2023-05-06	CUST005	Male	30	Beauty	2	50	100
...	...	...	...	...	...	...	...	...	...
995	996	2023-05-16	CUST996	Male	62	Clothing	1	50	50
996	997	2023-11-17	CUST997	Male	52	Beauty	3	30	90
997	998	2023-10-29	CUST998	Female	23	Beauty	4	25	100
998	999	2023-12-05	CUST999	Female	36	Electronics	3	50	150
999	1000	2023-04-12	CUST1000	Male	47	Electronics	4	30	120

1000 rows × 9 columns

Next steps: Generate code with df New interactive sheet

```
df.tail()  #least 5 rows
```

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount
995	996	2023-05-16	CUST996	Male	62	Clothing	1	50	50
996	997	2023-11-17	CUST997	Male	52	Beauty	3	30	90
997	998	2023-10-29	CUST998	Female	23	Beauty	4	25	100
998	999	2023-12-05	CUST999	Female	36	Electronics	3	50	150
999	1000	2023-04-12	CUST1000	Male	47	Electronics	4	30	120

```
#information
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   Transaction ID    1000 non-null   int64 
 1   Date              1000 non-null   object
 2   Customer ID       1000 non-null   object
 3   Gender            1000 non-null   object
 4   Age               1000 non-null   int64 
 5   Product Category  1000 non-null   object
 6   Quantity          1000 non-null   int64 
 7   Price per Unit    1000 non-null   int64 
 8   Total Amount      1000 non-null   int64 
dtypes: int64(5), object(4)
memory usage: 70.4+ KB
```

```
#describe
df.describe()
```

	Transaction ID	Age	Quantity	Price per Unit	Total Amount
count	1000.000000	1000.00000	1000.000000	1000.000000	1000.000000
mean	500.500000	41.39200	2.514000	179.890000	456.000000
std	288.819436	13.68143	1.132734	189.681356	559.997632
min	1.000000	18.00000	1.000000	25.000000	25.000000
25%	250.750000	29.00000	1.000000	30.000000	60.000000
50%	500.500000	42.00000	3.000000	50.000000	135.000000
75%	750.250000	53.00000	4.000000	300.000000	900.000000
max	1000.000000	64.00000	4.000000	500.000000	2000.000000

2.Data cleanikng and preprocessing

```
#checking misisng values
print(df.isnull())
```

```
     Transaction ID   Date  Customer ID  Gender    Age  Product Category  \
0             False  False        False   False  False             False   
1             False  False        False   False  False             False   
2             False  False        False   False  False             False   
3             False  False        False   False  False             False   
4             False  False        False   False  False             False   
..              ...    ...          ...     ...    ...               ...   
995           False  False        False   False  False             False   
996           False  False        False   False  False             False   
997           False  False        False   False  False             False   
998           False  False        False   False  False             False   
999           False  False        False   False  False             False   

     Quantity  Price per Unit  Total Amount  
0       False           False         False  
1       False           False         False  
2       False           False         False  
3       False           False         False  
4       False           False         False  
..        ...             ...           ...  
995     False           False         False  
996     False           False         False  
997     False           False         False  
998     False           False         False  
999     False           False         False  

[1000 rows x 9 columns]
```

```
print(df.isnull().sum(1))
```

```
0      0
1      0
2      0
3      0
4      0
```

```
       ..
995    0
996    0
997    0
998    0
999    0
Length: 1000, dtype: int64
```

```
#Remove duplicates
df = df.drop_duplicates()
```

```
df
```

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount
0	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
3	4	2023-05-21	CUST004	Male	37	Clothing	1	500	500
4	5	2023-05-06	CUST005	Male	30	Beauty	2	50	100
...	...	...	...	...	...	...	...	...	...
995	996	2023-05-16	CUST996	Male	62	Clothing	1	50	50
996	997	2023-11-17	CUST997	Male	52	Beauty	3	30	90
997	998	2023-10-29	CUST998	Female	23	Beauty	4	25	100
998	999	2023-12-05	CUST999	Female	36	Electronics	3	50	150
999	1000	2023-04-12	CUST1000	Male	47	Electronics	4	30	120

1000 rows × 9 columns

Next steps: Generate code with df New interactive sheet

```
#convert data column to datetime
df['Date'] = pd.to_datetime(df['Date'])
```

```
df
```

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount
0	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
3	4	2023-05-21	CUST004	Male	37	Clothing	1	500	500
4	5	2023-05-06	CUST005	Male	30	Beauty	2	50	100
...	...	...	...	...	...	...	...	...	...
995	996	2023-05-16	CUST996	Male	62	Clothing	1	50	50
996	997	2023-11-17	CUST997	Male	52	Beauty	3	30	90
997	998	2023-10-29	CUST998	Female	23	Beauty	4	25	100
998	999	2023-12-05	CUST999	Female	36	Electronics	3	50	150
999	1000	2023-04-12	CUST1000	Male	47	Electronics	4	30	120

1000 rows × 9 columns

Next steps: Generate code with df New interactive sheet

```
#ensure numerical col or values
df = df[(df['Quantity'] > 0) & (df['Price per Unit'] > 0)]
print("Cleaned dataset shape:",df.shape)
```

```
Cleaned dataset shape: (1000, 9)
```

```
df
```

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount
0	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
3	4	2023-05-21	CUST004	Male	37	Clothing	1	500	500
4	5	2023-05-06	CUST005	Male	30	Beauty	2	50	100
...	...	...	...	...	...	...	...	...	...
995	996	2023-05-16	CUST996	Male	62	Clothing	1	50	50
996	997	2023-11-17	CUST997	Male	52	Beauty	3	30	90
997	998	2023-10-29	CUST998	Female	23	Beauty	4	25	100
998	999	2023-12-05	CUST999	Female	36	Electronics	3	50	150
999	1000	2023-04-12	CUST1000	Male	47	Electronics	4	30	120

1000 rows × 9 columns

Next steps: Generate code with df New interactive sheet

3.Feature understanding

```
#Unique product categories
print("Product Categories:",df['Product Category'].unique())
```

```
Product Categories: ['Beauty' 'Clothing' 'Electronics']
```

```
print("Quantity:",df['Quantity'].unique())
```

```
Quantity: [3 2 1 4]
```

```
#Gender Distribution
print(df['Gender'].value_counts())
```

```
Gender
Female    510
Male      490
Name: count, dtype: int64
```

```
print(df['Product Category'].value_counts())
```

```
Product Category
Clothing       351
Electronics    342
Beauty         307
Name: count, dtype: int64
```

```
print(df['Customer ID'].value_counts())
```

```
Customer ID
CUST1000    1
CUST001     1
CUST002     1
CUST003     1
CUST004     1
           ..
CUST013     1
CUST012     1
CUST011     1
CUST010     1
CUST009     1
Name: count, Length: 1000, dtype: int64
```

4.Filtering ,Sorting & Subsetting

```
#Filter by a single category
electronics_df = df[df['Product Category'] == 'Electronics']
print(electronics_df)
```

```
     Transaction ID       Date Customer ID  Gender  Age Product Category  \
2                 3 2023-01-13     CUST003    Male   50      Electronics   
7                 8 2023-02-22     CUST008    Male   30      Electronics   
8                 9 2023-12-13     CUST009    Male   63      Electronics   
12               13 2023-08-05     CUST013    Male   22      Electronics   
14               15 2023-01-16     CUST015  Female   42      Electronics   
..              ...        ...         ...     ...  ...              ...   
988             989 2023-12-28     CUST989  Female   44      Electronics   
991             992 2023-08-21     CUST992  Female   57      Electronics   
992             993 2023-02-06     CUST993  Female   48      Electronics   
998             999 2023-12-05     CUST999  Female   36      Electronics   
999            1000 2023-04-12    CUST1000    Male   47      Electronics   

     Quantity  Price per Unit  Total Amount  
2           1              30            30  
7           4              25           100  
8           2             300           600  
12          3             500          1500  
14          4             500          2000  
..        ...             ...           ...  
988         1              25            25  
991         2              30            60  
992         3              50           150  
998         3              50           150  
999         4              30           120  

[342 rows x 9 columns]
```

```
# Filter rows where 'Product Category' is 'Electronics' and select the 'Quantity' column
electronics_quantity_df = df.loc[df['Product Category'] == 'Electronics', ['Quantity']]
print(electronics_quantity_df)
```

```
     Quantity
2           1
7           4
8           2
12          3
14          4
..        ...
988         1
991         2
```

```
992         3
998         3
999         4

[342 rows x 1 columns]
```

```
#sort by total amount
sorted = df.sort_values(by='Total Amount',ascending=False)
print(sorted)
```

```
     Transaction ID       Date Customer ID  Gender  Age Product Category  \
945             946 2023-05-08     CUST946    Male   62      Electronics   
71               72 2023-05-23     CUST072  Female   20      Electronics   
14               15 2023-01-16     CUST015  Female   42      Electronics   
576             577 2023-02-13     CUST577    Male   21           Beauty   
571             572 2023-04-20     CUST572    Male   31         Clothing   
..              ...        ...         ...     ...  ...              ...   
190             191 2023-10-18     CUST191    Male   64           Beauty   
43               44 2023-02-19     CUST044  Female   22         Clothing   
543             544 2023-12-23     CUST544  Female   27      Electronics   
988             989 2023-12-28     CUST989  Female   44      Electronics   
978             979 2023-01-02     CUST979  Female   19           Beauty   

     Quantity  Price per Unit  Total Amount  
945         4             500          2000  
71          4             500          2000  
14          4             500          2000  
576         4             500          2000  
571         4             500          2000  
..        ...             ...           ...  
190         1              25            25  
43          1              25            25  
543         1              25            25  
988         1              25            25  
978         1              25            25  

[1000 rows x 9 columns]
```

```
df
```

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount
0	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
3	4	2023-05-21	CUST004	Male	37	Clothing	1	500	500
4	5	2023-05-06	CUST005	Male	30	Beauty	2	50	100
...	...	...	...	...	...	...	...	...	...
995	996	2023-05-16	CUST996	Male	62	Clothing	1	50	50
996	997	2023-11-17	CUST997	Male	52	Beauty	3	30	90
997	998	2023-10-29	CUST998	Female	23	Beauty	4	25	100
998	999	2023-12-05	CUST999	Female	36	Electronics	3	50	150
999	1000	2023-04-12	CUST1000	Male	47	Electronics	4	30	120

1000 rows × 9 columns

Next steps: Generate code with df New interactive sheet

```
#subset selected columns
subset_df = df[['Date','Product Category','Quantity','Total Amount']]
print(subset_df.head())
```

```
         Date Product Category  Quantity  Total Amount
0 2023-11-24           Beauty         3           150
1 2023-02-27         Clothing         2          1000
2 2023-01-13      Electronics         1            30
3 2023-05-21         Clothing         1           500
4 2023-05-06           Beauty         2           100
```

5.Grouping Aggregration Analysis

```
#category summary
category_summary = df.groupby('Product Category').agg({'Quantity':'sum','Total Amount':'sum'})
print(category_summary)
```

```
                  Quantity  Total Amount
Product Category                        
Beauty                 771        143515
Clothing               894        155580
Electronics            849        156905
```

6.Pivot table table/Data Reshapping

```
pivot_table = pd.pivot_table(df,values='Total Amount',
                             index='Product Category',
                             columns='Gender',aggfunc='sum')
print(pivot_table)
```

```
Gender            Female   Male
Product Category               
Beauty             74830  68685
Clothing           81275  74305
Electronics        76735  80170
```

7.Descriptive stastical analysis

```
#mean & median
mean_quantity = df['Quantity'].mean()
median_quantity = df['Quantity'].median()
print("Mean Quantity:",mean_quantity)
print("Median Quantity:",median_quantity)
```

```
Mean Quantity: 2.514
Median Quantity: 3.0
```

```
#standard deviation
std_quantity = df['Quantity'].std()
print("Standard Deviation Quantity:",std_quantity)
```

```
Standard Deviation Quantity: 1.1327343409145405
```

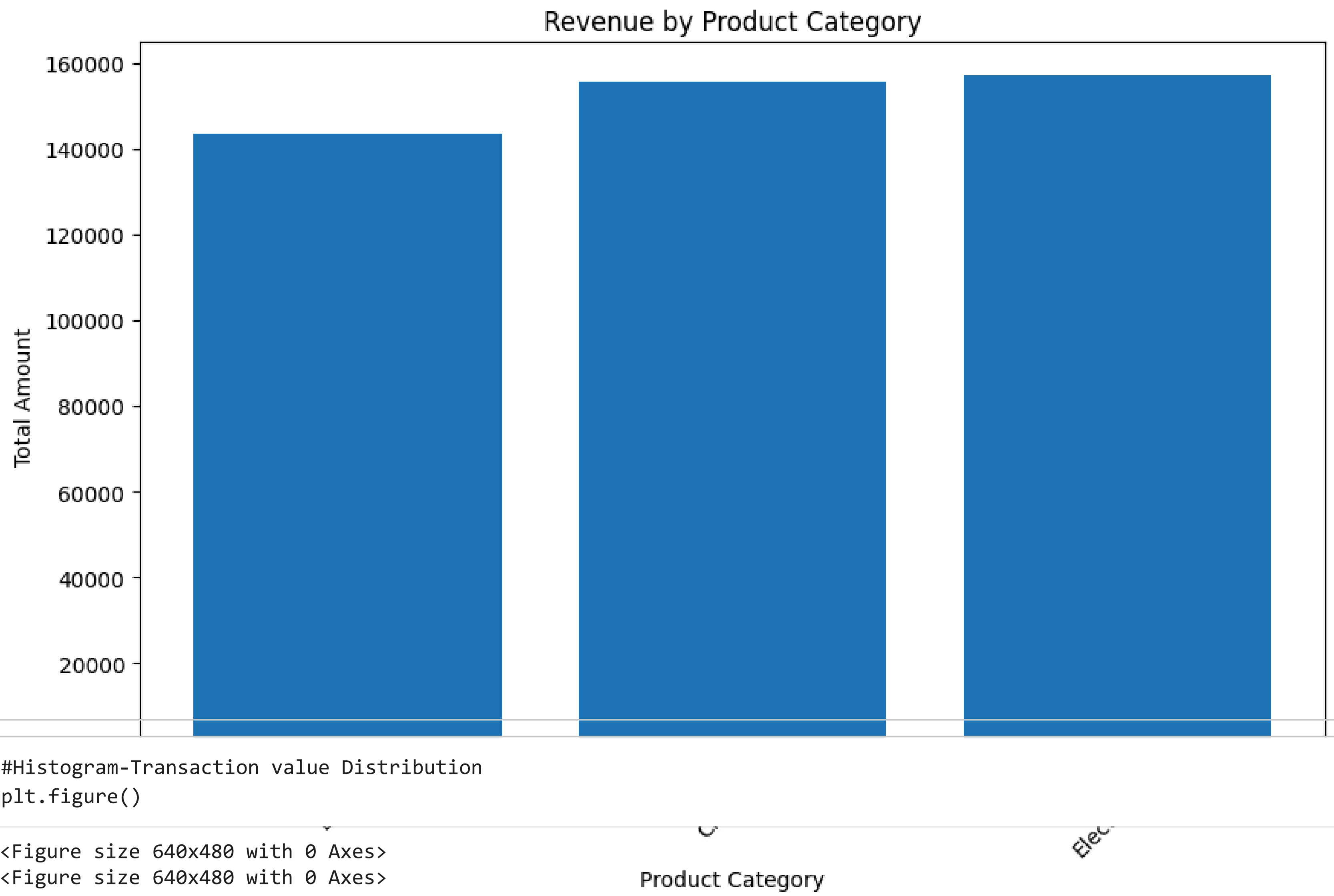
```
Start coding or generate with AI.
```

```
#Interquartile range(IQR)
Q1 = df['Quantity'].quantile(0.25)
Q3 = df['Quantity'].quantile(0.75)
IQR = Q3 - Q1
print("Interquartile Range:",IQR)
print("mean:",'mean_quantity')
print("median:",'median_quantiy')
```

```
Interquartile Range: 3.0
mean: mean_quantity
median: median_quantiy
```

8.Data visualization

```
#Bar Chart-Revenue by product category
plt.figure(figsize=(10,6))
plt.bar(category_summary.index,category_summary['Total Amount'])
plt.xlabel('Product Category')
plt.ylabel('Total Amount')
plt.title('Revenue by Product Category')
plt.xticks(rotation=45)
plt.show()
```



```
#Histogram-Transaction value Distribution
plt.figure()
```

```
<Figure size 640x480 with 0 Axes>
<Figure size 640x480 with 0 Axes>
```

9.Trend / pattern /outlier analysis

```
#trend
monthly_sales = df.groupby(df['Date'].dt.to_period('M'))[['Quantity', 'Total Amount']].sum()
```

```
df
```

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount
0	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
3	4	2023-05-21	CUST004	Male	37	Clothing	1	500	500
4	5	2023-05-06	CUST005	Male	30	Beauty	2	50	100
...	...	...	...	...	...	...	...	...	...
995	996	2023-05-16	CUST996	Male	62	Clothing	1	50	50
996	997	2023-11-17	CUST997	Male	52	Beauty	3	30	90
997	998	2023-10-29	CUST998	Female	23	Beauty	4	25	100
998	999	2023-12-05	CUST999	Female	36	Electronics	3	50	150
999	1000	2023-04-12	CUST1000	Male	47	Electronics	4	30	120

1000 rows × 9 columns

Next steps: Generate code with df New interactive sheet

```
#outlier detection using IQR

Q1 = df['Total Amount'].quantile(0.25)
Q3 = df['Total Amount'].quantile(0.75)
IQR = Q3 - Q1

high_value_sales = df[(df['Total Amount'] > Q3 + 1.5 * IQR)]
print("Hiogh Value Sales:")
print(high_value_sales)
```

```
Hiogh Value Sales:
Empty DataFrame
```

```
Columns: [Transaction ID, Date, Customer ID, Gender, Age, Product Category, Quantity, Price per Unit, Total Amount]
Index: []
```

```
#Revenue Contribution of High -value Transactions
high_value_revenue = high_value_sales['Total Amount'].sum()
total_revenue = df['Total Amount'].sum()
revenue_contribution = (high_value_revenue / total_revenue) * 100
print("Revenue Contribution of High-Value Transactions:",revenue_contribution)
print("Total Revenue:",total_revenue)
```

```
Revenue Contribution of High-Value Transactions: 0.0
Total Revenue: 456000
```

```
outliers =df[
    (df['Total Amount'] > Q1 + 1.5 * IQR) |
    (df['Total Amount'] < Q3 - 1.5 * IQR)
]
print("Number of outliers :",outliers.shape[0])
```

```
Number of outliers : 99
```

10.Business Insight Extraction

```
top_category = category_summary.sort_values(
    by='Total Amount',ascending=False
).head(1)

print("Top Revenue Generating Category:")
print(top_category)
```

```
Top Revenue Generating Category:
                  Quantity  Total Amount
Product Category
Electronics            849        156905
```

```
df
```

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount
0	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
3	4	2023-05-21	CUST004	Male	37	Clothing	1	500	500
4	5	2023-05-06	CUST005	Male	30	Beauty	2	50	100
...	...	...	...	...	...	...	...	...	...
995	996	2023-05-16	CUST996	Male	62	Clothing	1	50	50
996	997	2023-11-17	CUST997	Male	52	Beauty	3	30	90
997	998	2023-10-29	CUST998	Female	23	Beauty	4	25	100