In [3]:
```
import numpy as np
import pandas as pd
df=pd.read_csv("C:\\Users\\satya\\Downloads\\ecommerce_orders_10lakh_unclean_data.z
df
```

Out[3]:

	order_id	customer_id	order_date	ship_date	product_id	category	subcategory
0	1	88714.0	2022-01-01	2022-01-08	1007	Books	Sub3
1	2	81507.0	2022-01-01	2022-01-05	578	Electronics	Sub2
2	3	43523.0	2022-01-01	2022-01-14	1676	Fashion	Sub3
3	4	79947.0	2022-01-01	2022-01-12	1432	Toys	Sub2
4	5	95885.0	2022-01-01	2022-01-09	533	Electronics	Sub5
...	...	...	...	...	...	...	..
1039995	527016	27109.0	2023-08-01	2023-08-09	955	Toys	Sub3
1039996	889364	23185.0	2024-08-31	2024-09-08	976	Fashion	Sub1
1039997	802307	77837.0	2024-05-28	2024-06-10	1735	Electronics	Sub4
1039998	975109	73055.0	2024-12-03	2024-12-15	715	Fashion	Sub1
1039999	905539	94586.0	2024-09-18	2024-09-22	1680	Electronics	Sub2

1040000 rows × 13 columns

In [4]:
```
#Taking more information about the data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1040000 entries, 0 to 1039999
Data columns (total 13 columns):
 #   Column          Non-Null Count    Dtype  
---  ------          --------------    -----  
 0   order_id        1040000 non-null  int64  
 1   customer_id     1034768 non-null  float64
 2   order_date      1040000 non-null  object 
 3   ship_date       1019113 non-null  object 
 4   product_id      1040000 non-null  int64  
 5   category        1040000 non-null  object 
 6   subcategory     1040000 non-null  object 
 7   quantity        1040000 non-null  int64  
 8   price           1040000 non-null  float64
 9   payment_method  1040000 non-null  object 
 10  city            1029424 non-null  object 
 11  state           1040000 non-null  object 
 12  country         1040000 non-null  object 
dtypes: float64(2), int64(3), object(8)
memory usage: 103.1+ MB
```

In [5]:
```
#Total Null values in each column
df.isnull().sum()
```

Out[5]:
```
order_id             0
customer_id       5232
order_date           0
ship_date        20887
product_id           0
category             0
subcategory          0
quantity             0
price                0
payment_method       0
city             10576
state                0
country              0
dtype: int64
```

In [6]:
```
#First i will remove rows where customer_id is missing
df=df.dropna(subset=['customer_id'])
```

In [9]:
```
#now i will convert customer_id datatype float to int
df['customer_id']=df['customer_id'].astype(int)
```

In [35]:
```
#Now i will check duplicates in order_id column
df['order_id'].duplicated().sum()
```

Out[35]:
```
np.int64(39812)
```

In [10]:
```
#I will remove duplicate rows except the first occurence of each duplicate
df.drop_duplicates(subset=['order_id'],keep='first',inplace=True)
```

In [11]:
```
#Converting datatype of order_date and ship_date to datetime
df['order_date'] = pd.to_datetime(df['order_date'], format='%Y-%m-%d', errors='coer
```

```
df['ship_date'] = pd.to_datetime(df['ship_date'], format='%Y-%m-%d', errors='coerce
```

In [52]:
```
#creating a new column of delivery_time where delivery_days are present for each ro
df['delivery_time'] = (df['ship_date'] - df['order_date']).dt.days
```

In [11]:
```
#findinf median to fill missing ship_date values
median_delivery_time = df['delivery_time'].median()
print(median_delivery_time)
```

```
7.0
```

In [12]:
```
#Filling missing ship_date values
df['ship_date'] = df['ship_date'].fillna(df['order_date'] + pd.to_timedelta(7, unit
```

In [13]:
```
#Converting datatype of price in int
df['price']=df['price'].astype(int)
```

In [41]:
```
#Now checking the negative values in price and quantityy column
Negative_values=(df['quantity']<0).sum()
print(Negative_values)
Negative_values2=(df['price']<0).sum()
print(Negative_values2)
```

```
29516
29839
```

In [14]:
```
#Fixing Negative values in price and quantity column
df['quantity']=df['quantity'].abs()
df['price']=df['price'].abs()
```

In [15]:
```
#Filling Null values in city column so we have to find mode first
mode_city = df['city'].mode()
print(mode_city) # delhi is the mode
#filling null values to delhi
df['city'] = df['city'].fillna('Delhi')
```

```
0    Delhi
Name: city, dtype: object
```

In [16]:
```
#Finding unique values in category,payment_method,city
Unique_category=df['category'].unique()
print(Unique_category)
Unique_payment_method=df['payment_method'].unique()
print(Unique_payment_method)
Unique_city=df['city'].unique()
print(Unique_city)
```

```
['Books' 'Electronics' 'Fashion' 'Toys' 'Home' 'Toy' 'Fahsion' 'Hme'
 'Boks' 'Electrnics']
['UPI' 'Debit Card' 'COD' 'Net Banking' 'Wallet' 'Credit Card']
['Hyderabad' 'Delhi' 'Chennai' 'Lucknow' 'Bangalore' 'Kolkata' 'Mumbai']
```

In [17]:
```
#Fixing the incorrect values in category column
# Step 1: Define a mapping from incorrect to correct spellings
category_corrections = {
    'Boks': 'Books',
```

```
    'Electroncs': 'Electronics',
    'Fahsion': 'Fashion',
    'Fahion': 'Fashion',  # 'Fahion' also appears in the output
    'Hme': 'Home',
    'Toy': 'Toys',  # Decide if 'Toys' or 'Toy' should be standard
    'Electrnics': 'Electronics',  # in case this is another typo
    # Add any other common typos you find
}

# Step 2: Use pandas replace to correct these in the category column
df['category'] = df['category'].replace(category_corrections)
```

In [46]:
```
# we will remove state column as there are too much wrong values in that column
df = df.drop('state', axis=1)
```

In [47]:
```
#Creating a new column named as Total Value in which we will fill values by doing p
df['Total_Value']=(df['quantity']*df['price'])
```

In [56]:
```
#fixing datatype of delivery time column
df['delivery_time']=df['delivery_time'].astype(int)
```

In [57]:
```
# Save cleaned data to new CSV
df.to_csv("cleaned_ecommerce_orders.csv", index=False)
```